שאלות מכוונות במבני נתונים

מבוא

מהם הספרים הרלוונטיים ?
מבוא לאלגוריתמים

מהם הכרכים הרלוונטיים לקורס שלנו ?
והפרקים ?
שניהם

מה חדש או שונה בקורס זה ממה שלמדנו
כבר ב C ?
הקשר לתכנות זניח
מבוצע שימוש בקודים מנוונים

מהם מבני הנתונים עליהם דנים בקורס ?
מחסנית
תור
עצים שונים
רשימה מקושרת
ערימות
ועוד רבים מסוגים שונים..

מהם עקרונות הכתיבה בפסאודו קוד ?
קודם שורת הוראה ואז משתנים בסוגריים ?

מי משתמש במבנה נתונים בתעשייה ?
כולם
למשל מנוע חיפוש של גוגל

מה צפוי שיהיה במבחן של רחל ?
נוסחאות נסיגה
אין LCA
2:30, 3 שאלות כ"א 30 שאלה 1 10 נק'

תור ומחסנית

מהו תור ?
האיבר הראשון שנכנס הוא הראשון שיוצא

האם משנה כוון התנועה בתור ?
לא. אפשר להוציא את head ולהוסיף ל tail
או להיפך.

למה מאותחל ה next של איבר חדש לתור
ברשימה מקושרת ?
בעקרון ל NULL

מאיפה מכניסים ומאיפה מוציאים בתור, לפי
השיטות שנלמדו בכיתה ? איך זוכרים את
זה ?
הראשון תמיד בראש
לתור :
הכנסה – ל tail
הוצאה – מ head

מה הכוונה במימוש מבנה נתונים ע"י מודל
מידע מסוים (כגון רשימה מקושרת, מערך
וכו')

איך קוראים למצביע לאיבר אחרון ברשימה
מקושרת ?
Tail

לאן מוסיפים איבר בתור ?
לסוף הרשומה

מימוש מחסנית ע"י שני תורים

האם ניתן לממש מחסנית ע"י שני תורים ?
למה ?
כן
תור עזר ותור מחסן
או מימוש זיג זג

מהי הסיבוכיות של הוספת איבר ? למה ?
O(n)
כי צריך להעביר את כולם בין התורים

מה הסיבוכיות של הוצאת איבר ? למה ?
O(1)
רק צריך להוציא

סיבוכיות O זמן ומקום

מה זה O ? מה האות מסמלת ?
order זה
סיבוכיות
תכל"ס – סדר גודל, חסם עליון

מהו הסימון "O" ?
מסמנת חסם עליון ("עד") כדי קבוע לזמן הריצה
חסם אסימפטוטי עליון

מה נחשבת "פעולה" עבור חישוב זמן ריצה ?
שינוי ערך כלשהו.
ריצה "ריקה" על שורות קוד אינה מוסיפה זמן.

כיצד מסומן זמן הריצה הגרוע ? איך זוכרים את זה ?
כמו שאר זמני הריצה, אבל צריך לציין שמדובר בזמן הגרוע ביותר

האם בקורס יש לחשב סיבוכיות מדויקת, ולא "לעגל למעלה" כמו שעשינו ב C ?
בדר"כ סדר הגודל הוא מעוגל, אלא אם נאמר אחרת

מהי "תוחלת זמן הריצה" ?
Expected running time
$T(n)$?

האם הקצאת מקום נוסף היא תמיד דרך בניית מבנה חדש ? למה ?
לא
אפשר להוסיף נתון, כגון להוסיף שדה לרשומה מקושרת

טרנספורמי פורייה בדידים ?
מסמנת חסם תחתון עד כדי קבוע לזמן הריצה
חסם אסימפטוטי תחתון

מי יותר גדול, $lg(n!)$ או $4^{(lgn)}$? למה ?
$4^{(lgn)} = n^2$
$lg(n!) = n\ logn$

מי יותר גדול, 2^{2^n} או $(n+1)!$? למה ?
2^{2^n}
כי
$(n+1)! = O(n^n)$

מהו זמן מיזוג k רשימות שבהן סה"כ n איברים ? למה ?
$n \log k$
בכל שלב כולם (n) מתמזגים, ע"י $k / (2^i)$ רשימות

האם אפשר לומר שסיבוכיות מעבר על שני מערכים a,b כשידוע ש $a > b$ היא $O(a)$?
לא. חייב $O(a+b)$?

מהו הטריק של החזרות במערך ?
"טריק ההופעות"
אחרי שממיינים להוסיף לכל תא ערך נוסף של "מספר ההופעות".

האם
$\sqrt{n} = O(\log n)$? להיפך ? למה ?
לא
כן, כי נכון מעבר ל n מסוים

מהו a שמקיים
$\log^a (n) = O(\sqrt{n})$?
כל a קטן מאינסוף ?

מהם P ו NP בסיבוכיות ?
P סיבוכיות פולינומיאלית
NP סיבוכיות אקספוננציאלית

איך סופרים זמן ?
מספר פעולות לקלט הגרוע
אורך הקלט – מספר התאים בזכרון שתופס הקלט
כמה פעולות התוכנית עושה לקלט הנתון, במקרה הגרוע ביותר
פונקצית זמן
time : N > N
הכוונה לסדר גודל - O

מהו הניתוח הנוסף המתבקש לסיבוכיות ?
למה לא עושים את זה ?
לכל פעולה לוקח זמן שונה
כי אנחנו מדברים במושגי זמן של סדר גודל, אז הכפלה בקבוע לא תשנה

למה עובדים עם סדר גודל ולא סיבוכיות מדויקת ?
לפתור בעיית השוואה בין מחשבים שונים

איך סופרים זמן ?
מספר פעולות לקלט הגרוע
אורך הקלט – מספר התאים בזכרון שתופס הקלט
כמה פעולות התוכנית עושה לקלט הנתון, במקרה הגרוע ביותר
פונקצית זמן
time : N > N
הכוונה לסדר גודל - O

מהו הניתוח הנוסף המתבקש לסיבוכיות ?
למה לא עושים את זה ?
לכל פעולה לוקח זמן שונה
כי אנחנו מדברים במושגי זמן של סדר גודל, אז הכפלה בקבוע לא תשנה

למה עובדים עם סדר גודל ולא סיבוכיות מדויקת ?
לפתור בעיית השוואה בין מחשבים שונים

האם –
עבור פונקציה מונוטונית, ולפחות לינארית מתקיים
alfa <= 1
alfa * g(n) >=g(n *alfa)

כיצד בא לידי בסיבוכיות ביטוי אורך הפלט ?

חשב סיבוכיות זמן ריצה תוכנית המקבלת כקלט שני מספרים בינאריים ומחזירה מכפלה שלהם כפלט ?

האם סיבוכיות מקום תמיד קטנה מסיבוכיות זמן ?
אילן נוימן – כן
?

מהו זמן פולינומיאלי ?
$O(n^k)$

חסמים אסימפטוטיים

מהו חסם אסימפטוטי ?
פונקצייה סדר גודל החוסמת מלמעלה

מיהם החסמים האסמפטוטיים ? איך זוכרים ?
אומגא Ω – חסם תחתון. או**מ**גא חס**ם** תחתו**ם**
O – חסם עליון. כמו עיגול מכיל.
טטא Θ – חסם הדוק. בדיוק

כיצד מוכיחים את O של פונקציה כלשהי ?
משפט ההוכחה :
צ"ל : קיימים C>0, n0 כך שלכל n > no מתקיים :
$f(n) <= C * g(n)$

אילו C נבחר עבור הוכחות O ואילו עבור הוכחות אומגא ?
O – C גדולים
אומגא – C קטנים

מהי "טטא" ?
מסמנת שניתן "לכלוא" את כל זמני הריצה בין הזמן הקצר לזמן הארוך, עד כדי קבוע (כמכפלת פונקציה ידועה).
חסם הדוק אסימפטוטית.

מהו עקרון ההוכחה לכך שטטא מתקבלת ע"י בחירה "טהורה" של החזקה הגבוהה בלבד ?
מציאת c1, c2, n0 שמקיימים
$C1 * n^p <= f(n) <= c2 * n^p$

מהו T(n) ?
זמן מ Time

למה אנלוגי O, אומגה, וטטא ?
O : אם $f(n) = O(g(n))$ אז
$f(n) <= c\, g(n)$
Ω : אם $f(n) = \Omega(g(n))$ אז
$f(n) >= c\, g(n)$
Θ : אם $f(n) = \Theta(g(n))$ אז
$f(n) >= c\, g(n)$ וגם $f(n) <= c\, g(n)$

מהי יעילות אסימפטוטית ?
סדר גודל של זמן הריצה.
למה שואף זמן הריצה כשגודל הקלט שואף לאינסוף.

איך קובעים מהו "המקרה הממוצע" לחישוב סיבוכיות ?

מהו סימון "אומגה" ?
Ω

מהו עקרון ההוכחה להגדרות הסיבוכיות ?
מציאת C1, C2, n0 המתאימים להגדרות
הדרך הפשוטה להוכחה עבור פולינום – ביטוי C ע"י ערכים מוחלטים של שלושת המקדמים

האם
$n ^ {(1 + \sin n)} = O(n)$
? למה ?
לא. כי לא קיים c שעבורו ממקום מסוים
$c\, n >= n ^ {(1 + \sin n)}$
כי הפונקציה גם קטנה מ n ובהגדרה היא צריכה תמיד לחסום אותה ?

באיזה צד של המשוואה ה c ?
תמיד בצד של הפונקציה שבתוך הסוגריים של ה O

האם $n = O(n^2)$? ולהיפך ? למה ?
לא
כן. נבחר $c = 1$, $n0 = 1$

אם $f = O(g)$, איזה קשר אסימפטוטי נובע מכך ?
$g = \Omega(f)$

האם – פונק היא טטא של אחרת אמ"מ היא גם O וגם אומגא שלה ?
כן

האם – כל פונקציה חוסמת (טטא של) את עצמה ?
כן

עצי נסיגה

איך בונים עץ מנוסחת רקורסיה ?
$a T(f(n)) + g(n)$
a = מספר הענפים
$f(n)$ = העבודה בכל צומת, שינוי הפונקציה
$g(n)$ = במה מוחלף כל צומת שמוציאים לו ענפים

איך נראה העץ של
$T(n) = T(n/2) + n$? מהו הפתרון ?
שרשרת של nבאורך $\log n$
$n \log n$

בציור העץ, בכל רמה, האם $f(n)$ מושפעת ממספר הרמות שכבר ירדנו, או בכל צומת נעשית $f(n)$ עבודה בלי קשר לרמת הצומת ?
אין קשר לרמת הצומת

אינדוקציה

מהו שלד ההוכחה באינדוקציה ?
נכונות עבור בסיס
הנחה עבור n, והוכחה עבור n + 1

מה האתגר ?
למצוא מה בדיוק צריך להוכיח..

מה יכול לעזור כשנראה שההוכחה טריוויאלית ?
חלוקה למקרים

אילו אינדוקציות רלוונטיות לקורס זה ?
נוסחאות נסיגה
הוכחה שמיון עובד
הוכחת תכנון דינמי

איך בוחרים בסיס להוכחת נוסחת נסיגה באינדוקציה ?
בוחרים n כך ש n/b = 1 כדי שיתקבל T(1)

מהי הנחת האינדוקציה לנוסחת נסיגה ?
דוגמא :
הנחה: עבור $k<n$ מתקיים $T(k) \ge c\log^2 k$

איך מוכיחים נוסחה באינדוקציה ?
בוחרים בסיס n=b ורושמים את ביטויו ע"פ T(1) | רושמים "הנחה : נניח T(n)=___ עבור $k<n$. נוכיח עבור n :" | מתחילים משמאל של הדף | כותבים " = T(n)" | כותבים מה הוא שווה לפי הנוסחא הרגילה | במקום ה- T הקטן יותר מציבים את הנחת האינדוקציה (עם c) עבור הביטוי שבתוך ה T, ורושמים = ואת הביטוי החדש מימין | רושמים מימין = או => או <= לפי מה שמנסים להראות – teta omega O, ומעליו סימן שאלה.| בצד ימין רושמים את הנחת האינדוקציה המקורית עם c | מפתחים ומפשטים | מראים c ו n0 שמקיימים את אי השוויון | ה- C וה- N0 צריכים להיות טובים גם לבסיס וגם להוכחה הגדולה (עבור אותם C ו- N0) | אתה אמור למצוא C ו- N0 שמקיימים את הנחת האינדוקציה. אם אתה מוצא הצלחת ואם אין כאלה אז לא הוכחת

נוסחאות נסיגה

מהי נוסחת נסיגה ?
נוסחה שמגדירה סדרת איברים, כך שכל איבר בסדרה מוגדר רקורסיבית בעזרת האיברים הקודמים בה, פרט למספר סופי של איברים ראשונים, שמהווים את תנאי ההתחלה.
נוסחת ה T לביטוי הזמן הדרוש עבור שגרה רקורסיבית נתונה

איך מגיעים לנוסחת ה T מהפסאודו קוד ?
T(n) = a T(b) + C
T(n) = a T (n/b) + f(n)
a מספר הקריאות הרקורסיביות בתוך כל מופע של הלולאה
b גודל כל תת בעיה הנשלחות לקריאות הרקורסיביות, בעקרון פונק' של n
C העבודה הנעשית בכל קריאה. כלל העצירה ?

האם C מתייחס לסך כל העבודה ברמה של העץ או לעבודה הנעשית בכל עלה ?

האם תמיד יש לפתור נוסחת רקורסיה ע"י עץ ?
זו הדרך המסודרת
?

מהי נוסחת נסיגה ?

איך מגיעים מנוסחת ה T לביטוי הסופי של הסיבוכיות ?

מה היחס בין חסמי זכרון לזמן ריצה ?
זכרון תמיד חוסם מלמטה את זמן הריצה, בגלל איתחול המערכים

מהי נוסחת נסיגה ?

איך פותרים משוואת רקורסיה ?

מה ההבדל בין משוואת רקורסיה לנוסחת נסיגה ?
אין

מהן השיטות לפתירת נוסחאות נסיגה ? מהן בגדול ?
שיטת ההצבה – ניחוש חסם עליון והוכחתו באינדוקציה. שיטה מתמטית.
שיטת האיטרציה – הכי אינטואיטיבית.
פריסת העץ וספירת פעולות ?
משפט האב – יכול היה להיות באותה מידה "שיטת האם.." שיטה מאוד טכנית ופשוטה.

מהי שיטת האיטרציה ?
שרטוט "עץ עבודה". חלוקת הפונק' הרקורסיביות וכמה עבודה נעשית בכל שלב

מה "מסוכן" בשיטת ההצבה ?
היא נפתרת ל O ולא לטטא, אז יכולה להיפתר לפתרון לא מספיק מדויק. חסםלא "הדוק".

האם מותר להוכיח במבחן בשיטת האיטרציה ?
לא.
אפשר למצוא בעזרתה את התשובה ואז להשתמש בשיטות האחרות לפורמליות

מה המשמעות של פעמיים T בנוסחת הנסיגה ?

מהי החלפת משתנים לצורך פתרון משוואת נסיגה ?
כמו שיטת ההצבה לפתרון אינטגרלים..

מדוע בחלק מהמשוואות רשום C ובחלק f(n) ?
נוסחת האב מתייחסת רק ל f(n) ?

איך פותרים נוסחה מהצורה
T(n) = log T(n-1) + 1
? מה הפתרון של זו ?
teta(1)

מהם ה "צעדים" האפשריים להוכחה באינדוקציה ? איך בוחרים בינהם ?
מתקיים עבור n-1 נוכיח עבור n
מתקיים עבור n נוכיח עבור n+1
מתקיים עבור k<n נראה עבור k > n ?

שיטת האב

מהם התנאים לשיטת האב ?
T(n) = a T (n/b) + f(n)
a >= 1, b > 1, f(n) פונ' חיובית אסימפטוטית

מה אומר משפט האב ?
בתנאים שהוגדרו, ניתן לחסום את T כך :
1 = אם קיים eps > 0 כך ש

$$f(n) = O(n^{\log_b a-\varepsilon})$$

אז

$$T(n) = \Theta(n^{\log_b a})$$

2 = אם

$$f(n) = \Theta(n^{\log_b a})$$

אז

$$T(n) = \Theta(n^{\log_b a} \cdot \log n)$$

3 = אם

= אם קיים תנאי רגולריות וגם קיים eps > 0 כך ש

$$f(n) = \Omega(n^{\log_b a+\varepsilon})$$

אז

$$T(n) = \Theta(f(n))$$

מהו תנאי הרגולריות ?
af(n/b) <= cf(n)
for some constant c < 1 and all sufficiently large n.

מהי נוסחת האב ?

מהו משפט האב ?
Master theory
בהנתן נוסחת נסיגה מהצורה..
?
נותן שלושה מצבים לפתרון נוסחת נסיגה :
עלים כבדים
משקל שווה בכל רמה
ראש כבד

מהם המקרים בהם לא ניתן להשתמש בנוסחת האב ?
א) a אינו קבוע ב) n מול n logn
(פולינומיאליות..) ג) הצורה c = n / lg n
ד)a<1 ה)f(n) אינה חיובית ו)מקרה 3 ללא תנאי הרגולריות

There does not exist a *c* < *1* such that

$$2\frac{2}{n} \le c\frac{1}{n}$$

מתי יש אומגא ומתי O בשיטת האב ?
case 1 = O
case 3 = omega

האם יתכן פתרון עם שיטת האב עבור
f(n) = 2^n ? f(n) = 1 / n
כן
כן

מה בדרך שוכחים עבור מקרה 2 ?
שהלוג הרגיל מוכפל ב log n

מתי ולמה חשוב לציין ש T(1) = teta(1) ?
לפני שימוש בנוסחת האב ?

מהו סדר פתרון משוואה רקורסיבית ? למה לא תמיד מיד להשתמש במשפט האב ?

מה ההבדל בין פער פולינומיאלי לבין פער לוגריתמי ? מה הקשר למשפט האב ?
פער לוגריתמי : n לעומת n log n
פער פולינומיאלי : n לעומת n^2
במשפט האב צריך פער פולינומיאלי. איפה ?
ז"א, אי אפשר להשתמש כשיש n מול nlogn
= non-polynomial difference between f(n) and $n^{\log_b a}$
אסור להשתמש במשפט האב כשהחזקה של n קטנה בהפרש לוג מ f ?

= *a*<1 cannot have less than one sub problem

האם ייתכן מצב שעבור שתי פונקציות f(n) יתקבל אותו פתרון למשוואת הרקורסיה ?
כן. למעשה יש טווח פונקציות עבור כל פתרון, שנובע מנוסחת האב.

איך יודעים מהי הסיבוכיות הנמוכה ביותר האפשרית עבור כל f(n) ? מה המשמעות של נתון זה ?
מציבים f(n) = 1

האם תמיד T(1) = teta (1) ?
כן. זו ההנחה אלא אם נאמר אחרת

גרפים

גרפים 1

מהן שכבות הגרף ?
שכבת מבנה – שלד
שכבת תוכן – תוכן הצמתים

למה V הם קודקודיים ו E קשתות ? איך זוכרים את זה ?
ה V מסמל חץ שמצביע על נקודה
ה E הוא עיוות של arch

הנתיב הקצר ביותר ?

דרגות, גרף מכוון \ לא מכוון, איזומורפיזם, קשירות, גרפים דו צדדיים

שימושים בנושא גרפים :
מפות כבישים, מולקולות וכימיה (קשרים כפולים, איזומרים), מבנה ארגוני, קשרים בין חלבונים ביולוגיים, מידול צ'יפים, תרשימי זרימה של תהליכים ,

V ?
Vertices
צמתים, קודקודים

E ?
Edge
קשתות או צלעות או ענפים

מהם כל הסימולים ב?

מה פירושם

מהו זוג סדור ? מהו זוג לא סדור?
מספרים עוקבים

מהו גרף מכוון \ לא מכוון?
קשתות עם חץ כוון
קשתות בלי כוון

גרף מכוון באנגלית?
Digraph
Directed graph

מהי דרגה של צומת ? נכנסת ? יוצאת ?
מס' הקשתות המחוברות אליה
או נכנסות
או יוצאות

Degree
דרגה

בכמה מוסיפה לולאה לדרגתה של צומת בגרף מכוון ? ובלא מכוון ? למה?
2,אחת יוצאת ואחת נכנסת
2

קו בלי חץ?
כמו שני חצים דו כיווניים

מהו גרף דליל ?
גרף שאין בו מספר רב של קשתות. בצורה רשמית גרף דליל מוגדר כגרף שבו מספר הקשתות קטן ממספר הקודקודים בריבוע.

מהו גרף צפוף ?
גרף שאינו דליל. גרף צפוף הוא גרף שיש בו מספר רב של קשתות. בצורה רשמית גרף צפוף מוגדר כגרף שבו מספר הקשתות גדול ממספר הקודקודים בריבוע.

גרפים 2

M(U)
קבוצת השכנים
קבוצת כל ה V שיש להם קשת לU

מהו שכן של צומת? U
צומת אחר שיש לו קשת עםU

האם ייתכן שכן גם בגרף מכוון וגם בלא?
כן.
אבל הגדרות שונות.
שכן של U בגרף מכוון הוא מי שיש קשת מ
Uאל השכן, אבל לא להיפך.

למה : בגרף לא מכוון מס' הצמתים בעלי דרגה אי זוגית הוא זוגי?
כי 1 ועוד אי זוגי שווה זוגי
וגם מפיתוח הנוסחה

שלושת הנוסחאות מההרצאה:
נמצא בצילום ההרצאה שברגר שלח אלי

האם : בגרף לא מכוון סכום כל הדרגות הוא זוגי ? למה?
כן

סוגי גרפים ? הגדרה?
גרף מלא – בלתי מכוון שכל קודקודיו צמודים
מעגל
גלגל מגודלח
קוביה – תלת מימדי
פשוט \ לא
מכוון \ לא
דו"צ \ לא
תת גרף – מוכל בגרף יותר גדול
גרף משלים

מתי זוג צמתים יסומן בסוגריים מסולסלים ומתי בעגולים?

מהו פסאודו-גרף?
גרף שמותר שיהיו בו לולאות וגם מותר שתהיה יותר מקשת אחת בין שני צמתים (או מצומת אחד לצומת שני עבור גרף מכוון).

מהו תת גרף ? מהו תת גרף שלם?

מהו גרף משלים?
גרף פשוט שיש בו את הקשתות שאין בגרף פשוט בעל אותם קדקודים

האם לכל גרף פשוט קיים גרף משלים פשוט ?

מהי עוצמה של גרף?

מהו גרף? Kn
גרף לא מכוון בעל קודקודים ובין כל שני קודקודים ישנה קשת ?

מהו גרף מלא? Kn

מהו מעגל אוילר?
מסלול שעובר בכל הקודקודים מבלי לחזור על קשתות יותר מפעם אחת

מהו מעגל המילטון?
מסלול העובר בכל הקודקודים רק פעם אחת ?

מהו תת גרף ?
גרף המכיל חלק מהנקודות וחלק מהקשתות של הגרף המקורי

מהי "למת לחיצת הידיים" ? מהו עקרון ההוכחה שלה ?
Sigma degree(v) = 2|E|
כמות הצמתים הוא

מהו עקרון ההוכחה לחידת לחיצת הידיים הזוגית ?
אם מספר הקודקודים זוגי – אז גם המכפלה זוגית
אם מספר הקודקודים א זוגי – דרגת כל קודקוד זוגית

מהו מולטי גרף ?
גרף בלתי מכוון שיכול להכיל קשתות מרובות בין קודקודים ולולאות עצמיות

מהי מטריצת סמיכויות ?
מטריצה ריבועית שבה ערכים בינארים.
עבור שני צמתים שבניהם קשת נרשום 1, אחרת 0

האם שמיניה היא מעגל פשוט ?
לא

מהו מסלול המילטוני ?
עובר בכל צומת בגרף בדיוק פעם אחת

גרף מלא, קשיר, פשוט ומסלול פשוט

מהו גרף פשוט ?
ללא לולאות
אין בו יותר מקשת אחת באותו כוון (או ללא כוון) בין שני צמתים.
בקיצור : ללא לולאות וללא כפילויות

מהו מסלול פשוט ?
כל הצמתים לאורכו שונים

האם גרף פשוט יכול להיות מכוון ? ולא מכוון ?
יכול להיות גם וגם

האם גרף פשוט יכול להכיל 2 קשתות בכוונים מנוגדים בין אותם צמתים?
גרף פשוט מכוון יכול

האם תתכן נקודה בודדת בגרף פשוט?
בהחלט

האם בגרף פשוט ייתכנו שני חיצים בין שני צמתים, כל אחד בכוון הפוך?
גרף פשוט מכוון יכול

מהו מסלול פשוט ?
שכל הקודקודים בו שונים

האם "שמיניה" מארבעה קודקודים נחשבת למעגל פשוט ?
כן ?

מהו גרף קשיר ?
בין כל זוג קודקודים קיים מסלול

איזומורפיזם

מהם גרפים איזומורפים ?
ניתנים לקישור מקביל או כימות זה לזה

האם ייתכנו שני גרפי מעגל שונים בגודל 4 ?
איך ?
כן.
נקרא איזומורפיזם

מהי ההגדרה לגרפים שווים?
צמתים וקשתות שווים

מה ההבדל בין שווה לשקול?
זהה לעומת ששווה משמעות?

דוגמא לאיזומורפיזם למרובע (גרף מעגל מדרגה 4?)
שמיניה על בסיס 4 נקודות

מהם איזומורפים לגרף לא מכוון ? ומכוון?
2גרפים פשוטים נקראים איזומורפים
אם קיימת פונ' חע"ל f מ V1 ל V2 כך ש :
a ו b שכנים ב G1 אמ"מ
f(a) ו f(b) שכנים בG2
כלומר f משמרת שכנויות

מהם התכונות בין גרפים איזומורפיזם?
מספר צמתים שווה
מספר קשתות שווה
התפלגות דרגות שווה

כמה פונקציות חע"ל אפשריות בין גרפים בגודל n ? למה?
n!
אפשרויות סידור .
לנקודה הראשונה אפשריות n צמתים בגרף השני, א"כ n-1 וכו.'

DFS
Depth first search
חיפוש לעומק תחילה
אלגוריתם שמנסה למצוא איזומורפיזם ע"י בדיקת כל האפשרויות.

מהו תהליך בדיקת איזומורפיזם בין שני גרפים?
כמות צמתים
כמות קשתות.
התפלגות דרגות
מציאת דרגות המעגלים בגרפים – המסלול הקצר מנקודה ועד לאותה נקודה חזרה
חיפוש שכנויות "שלא מסתדרות" : מציאת נקודה בגרף אחד שאין לה נקודה בגרף השני ששכנויותיה הן מאותה דרגה.

הדרך המפורטת לכתוב זאת היא להגיד מהן הנקודות מהגרף השני שהן מאותה דרגה, ואז להראות לכל נקודה אם ייתכן, ולהמשיך עד להפלת כל הנקודות.

מהם השימושים של איזומורפיזם?
למשל, כימיה – השוואה בין מולקולות שמיוצגות עי גרפים

כיצד ניתן לתאר גרף קובייה באופן דו מימדי ?
ריבוע בתוך ריבוע עם חיבור בין קודקודים קרובים בין הריבועים

מסלול

מהו מסלול בגרף לא מכוון?
מסלול מצומת u לצומת v בגרף פשוט לא מכוון הוא סדרה של זוגות סדורים שבינהם קשתות

איפה עוד ראינו מסלול ? באיזה הקשר?
ברלציות

מהו אורך של מסלול?
מספר הקשתות שמכיל

האם ייתכן יותר ממסלול אחד בין צמתים?
כן

כמה מסלולים קיימים בין שתי נקודות?
אם קיים מסלול אחד, קיימים אינסוף מסלולים.

מהו מסלול פשוט?
שכל הקשתות שונות

מהו מעגל?
מסלול שחוזר לנקודת היציאה

האם הלוך חזור הוא מעגל ? למה ?
כן, כי יוצא וחוזר לאותה נקודה

מה ההבדל בין לולאה לנקודה?
אורך המסלול הוא 1 לעומת אפס

האם יש מסלול ? a מה אורכו?
כן
אפס

קשירות

מתי גרף לא מכוון נקרא "קשיר? "
אם קיים מסלול בין כל שני צמתים

האם ייתכן גרף לא קשיר שבו לכל נקודה יש שכן?
כן. המורכב משני גרפים שאין קשת בינהם.

Gהוא גרף לא מכוון.
Rמכיל את כל הזוגות שקיים בינהם מסלול.
האם R שקילות?
כן. מקיים את שלושת התכונות

מהן מחלקות השקילות של? R
תתי גרפים קשירים שנקראים "רכיבי קשירות."

מהם רכיבי קשירות?

מהו תת גרף?

מה זה קשיר חזק?
גרף מכוון נקרא "קשיר חזק" אם לכל שני צמתים a ו b קיים מסלול מ a ל b וגם מ b ל a

מהו "קשיר חלש? "
לא לומדים את זה

איזו רלציית R מקיימת שקילות בגרף מכוון?
אם לכל שתי נקודות יש מסלול הלוך וחזור

רכיבי קשירות חזקה?

איך מחלקות השקילות

מה אפשר לומר על נקודה שאף קו לא מגיע אליה בגרף מכוון?
שהוא מחלקת שקילות

האם : אם יש בגרף רכיב שקילות אחד אז הוא קשיר ? למה?
כן

האם : אם גרף לא מכוון הוא קשיר אז יש בו רכיב קשירות אחד ? למה?
כן

שימושים של רכיבי קשירות?
רשת מחשבים.
האם קשיר = האם כל המחשבים מחוברים זה לזה?

האם מולקולה היא תמיד גרף קשיר ? למה ?

מהגדרתה?

איך ממירים מגרף מכוון ללא ולהיפך ?
קשתות חד כיווניות נמחקות
קשתות דו כווניות הופכות ללא מכוונות
?

גרפים דו צדדיים

מהו גרף דו צדדי?
הגדרה : גרף G=(V,E) אם ניתן לחלק את V לשתי קבוצות זרות ולא ריקות, כך שאם קיימת קשת אז קצוותיה מקבוצות שונות.
לא פורמלי:
ניתן לחלק את צמתי הגרף לשתי קבוצות.
ואז – אין קשתות בתוך כל קבוצה אלא רק בינהן.

האם גרף דו צדדי הוא לגרף מכוון או לא?
לשניהם

האם ניתן לחלק כל גרף כך שיהפוך לדו"צ ?
לא – לדוגמא, משולש מלא

האם גרף דו"צ יכול להכיל קודקודים בודדים (לא מקושרים) ?
תלוי בהגדרתו.
בעקרון כן

מהו סימון לגרף דו"צ ?
G=(V1, V2, E)

איך מראים אם גרף הוא דו"צ?
מראים חלוקה לשתי קבוצות זרות שעונות להגדרה

האם מעגל יכול להיות דו"צ ? דוגמא?
כן.
מרובע.

האם כל מעגל יכול להיות דו"צ ? למה?
רק אם מס' הצמתים זוגי
לפי עקרון שובך היונים : בכל חלוקה לשתי קבוצות זרות תהיה קבוצה עם 2 צמתים, ובהכרח יש בינהם קשת

האם הגרף הריק הוא דו"צ ? למה?
כן

טיפ : איך אפשר לדעת אם גרף מסובך הוא דו"צ?
אם הגרף מכיל מעגל אי זוגי אז זה מפיל את הדו"ציות

מהו גרף דו צדדי שלם?
ניתן לחלק את V לשתי תתי קבוצות לא ריקות וזרות, כך שקיימת קשת אמ"מ שני צמתים שייכים לקבוצות שונות.

האם בגרף דו צדדי שלם מכל צומת בקבוצה אחת יש קשת לכל הנקודות בקבוצה הזרה השנייה?
כן

מהו מספר הקשתות בגרף לא מכוון דו"צ מלא שמספר הצמתים בשתי הקבוצות הוא m ? n ובגרף מכוון?
n * m
n * m * 2

האם : לכל גרף דו צדדי יש חלוקה אחת בלבד לרכיבי הקשירות ?
כן ?

מהו תנאי מספיק והכרחי לכך שגרף לא מכוון הוא דו צדדי ?
שאין בו מעגלים באורך אי זוגי

אלגוריתמים על גרפים

מהם סוגי הסריקה העיקריים על גרפים ?
סריקה לעומק לעומת לרוחב

מהי סריקה לרוחב ?
מגלה כל פעם את כל הקודקודים הנמצאים
באותו מרחק מקודקוד המקור

מהי שיטת "הפרד ומשול" עבור גרפים ?

מהו אלגוריתם SCC ?

מתי אפשר לומר שסיבוכיות
$O(|V|+|E|) = O(|V|) = O(|E|)$
?
למשל כאשר ידוע שהגרף הוא עץ

האם לגרף דו צדדי יש רק חלוקה אחת לדו
צדדיות ?
כן ?

מהם הטריקים לפתרון שאלות במבחן ?
זרימה -
אסתי - הכפלת הצמתים והוספת s ו t
אסתי אקסי – כמו אסתי אבל עבוד שתי
קבוצות X ו Y
טריק קשצומת – פיצול צמתים לצומתי
כניסה ויציאה
הטריק של מספר קומבינציות קבוע
הטריק של פיצול קשתות לצמתים והפעלת
bfs
הטריק של החתכים המינימליים
ווריאציות על אלג' מוכרים
עצים פורשים –
שינוי משקלי הקשתות ליצירת עדיפויות
אלגוריתם הקיבוץ
טריק יער רכיבי הקשירות
אלגוריתם למציאת שורש בעץ מכוון
הטריק של הפיכת כוון הקשתות
הטריק של הוספת צומת חיצוני לקבוצת
צמתים מסוימת
גרף העל - אלגוריתם מציאת רכיבי קשירות

מימוש גרפים

מהן הדרכים לייצוג גרפים בקוד ? מה עקרון כל ייצוג ?
מטריצת שכנויות – מטריצה בינארית שבה 1 מייצג קשת בין שני צמתים ו 0 מייצג "אין קשת"
רשימה מקושרת – מכל צומת משתרשרים הצמתים השכנים – שיש אליהם קשת ?

בהינתן ייצוג של גרף מכוון על ידי רשימות סמיכות, כמה זמן יידרש לחישוב דרגת היציאה של כל קודקוד ? למה ?
O(VE)
יש לעבור על כל קודקוד ועבורו לספור את כל הקשתות

בהינתן ייצוג של גרף מכוון על ידי רשימות סמיכות, כמה זמן יידרש לחישוב דרגת הכניסה של כל קודקוד ?
O(VE)
סריקה אחת של רשימת הסמיכויות

מתי כדאי להשתמש באיזה יישום ?
גרפים דלילים כדאי לייצג באמצעות רשימות סמיכות החסכוניות במקום.
גרפים צפופים כדאי לייצג באמצעות מטריצת סמיכויות .

BFS – סריקה לרוחב

BFS ?
Breadth-First-Search
אלגוריתם המוצא את המסלולים הקצרים בגרף לא ממושקל, מכוון או לא מכוון .

מהו האלגוריתם היעיל למציאת מסלול קצר מצומת לצומת בגרף מכוון ?
BFS

מהו רעיון האלגוריתם ?
הליכה על הגרף מצומת התחלה נתון באופן מחזורי
עי בחירת חוקיות כלשהי לסדר קבוע של הצומת הבאה לבדיקה
בכל צומת שעוברים נספרים מספר הצעדים שבוצעו מ S
כל צומת אליו הגענו כבר מקבל תג 1, שאחר שאותחל בתחילה לאפס
אם הצומת הבא מסומן 1 אז מתעלמים ממנו
בכל צומת אליו מגיעים מוחקים את כל הקשתות הנכנסות אליו ב O(1) ?

מהי השיטה ?
האלגוריתם משתמש בתור כדי לקבוע מהו הצומת הבא בו הוא עומד לבקר. בכל פעם שהוא מבקר בצומת הוא מסמן אותו ככזה שנבדק, ואז בודק את כל הקשתות שיוצאות ממנו. אם קשת מובילה לצומת שטרם נבדק, צומת זה מתווסף לתור. בדרך זו מובטח כי האלגוריתם יסרוק את הצמתים בסדר שנקבע על פי מרחקם מהצומת ההתחלתי (כי צומת שנכנס לתור יצא ממנו רק לאחר שכל הצמתים שהיו בו קודם יצאו).

מהו פלט האלגוריתם ?
עץ שמקיים את התכונה שהמסלול משורש העץ לכל אחד מהצמתים הוא המסלול בעל מספר הצלעות הנמוך ביותר בגרף המקורי.

למה נהוג להשוות BFS ?
גל מתפשט מהצומת ההתחלתית,
או, תכנון דינאמי למספר הצעדים הקצר מצומת ההתחלה לכל צומת אחרת

האם : המסלול הקצר בין כל שתי נקודות אמצע על מסלול קצר משתי נקודות קצה, עובר על אותו מסלול ארוך ? למה ?
כן
אינדוקציה ?

מהו מימוש יעיל ל BFS ?
תור queue מערך
?

מהו d(s) ?
אורך המסלול מ s ?

מהו d(s,v) ?
אורך המסלול מ s ל v ?

מהי סיבוכיות BFS ? למה ?
O (|V| +|E|)
על כל צומת עושים פעולה אחת ועל כל קשת פעולה אחת

האם BFS עובד עבור גרף לא מכוון ? מה ההבדל ?
כן
עבור גרף מכוון יש לדעת שורש אחרת ידלג בין רכיבי קשירות
שתי פעולות במקום אחת בכל קשת ?

מה מייצג n ?
|V|

מהי סיבוכיות האלגוריתם ? למה ?
סיבוכיות זמן ומקום: O(|E|+|V|).
לאחר האיתחול, בו צובעים את כל הקודקודים בלבן, לא נצבע אף קודקוד בלבן ולכן כל קודקוד נכנס ויוצא מהתור לכל היותר פעם אחת.
כל פעולת הכנסה והוצאה מהתור היא O(1)
לכן סה"כ הזמן המוקדש לתור הוא O(V).
הזמן הכולל הלוקח לסריקת הקשתות (בהנחה שהגף מוצג ברשמית סמיכויות) הוא כסכום אורכי הקשתות, כלומר O(E).
זמן הריצה הכולל של האלגוריתם הוא לינארי כגודל הייצוג של הגרף, כלומר O(E+V).

אם במקרה הגרוע |E| = |V|^2 אז למה בעצם האלגוריתם נחשב לינארי ?
הכוונה ללינארי ביחס למספר הצמתים ועוד הקשתות הנתון

האלגוריתם פועל הן על גרפים מכוונים או לא ?
גם וגם

איך נראה הקוד ?

```
for each vertex u (who is not s - the source vertex)
    color(u) = white
        d(u) = inf
color(s) = gray
d(s) =0
Enqueue(Q,s)
while Q not empty
    u = head(Q)
    for each v neighbor of u
        if color(v) = white then
            d(v) = d(u)+1
            Enqueue(Q,v)
    Dequeue(Q)
    color(u) = black
end while
```

למה נקרא גם "גל מתפשט" ?
כי עובר על הצמתים לפי מרחקם מ s

מתי אפשר לומר שסיבוכיות האלגוריתם O(|E|) ? למה ?
כאשר הגרף קשיר.
קיימות לכל הפחות |V|-1 קשתות

נתון עץ לא מכוון. מצא אלגוריתם יעיל המוצא את שני הצמתים שהמרחק המינימלי בינהם הוא המרבי ?
שתי ריצות bfs. אחת מצומת שרירותי, השניה מהצומת הרחוק ביותר.

נתון גרף עם קשתות אדומות או שחורות. מצא אלגוריתם לינארי למסלולים קצרים מ s עם מעבר בלפחות קשת שחורה אחת ? ובדיוק ?
ווריאציה על bfs – שני סימונים לכל צומת: עם או בלי התנאי. סריקה עד שמגיעים לכל הצמתים עם התנאי
או – בניית גרף שקול עם צמתי עזר עבור אי עמידה בתנאים

מהו משפט "מסלול חוצה חתך" ?

דייקסטרה

מהי דייקסטרה ?
פותר את בעיית מציאת המסלול הזול ביותר מנקודה לשאר הנקודות, בגרף מכוון או לא, בעל משקלים אי שליליים.

מהו עקרון אלגוריתם דקסטרה ?
כמו BFS אבל עם priority queue או ערימת מינימום ?

מהי הסיבוכיות של דקסטרה ? למה ?
תלוי במימוש התור
הכי יעיל בערימת מינימום פיבונאצ'י ?
? $O(|V| \lg |V| + |E|)$
בגלל פעולות על ערימה – הוצאה, החלפה, מציאת מינימום

אם משתמשים בערימה בינארית זמן הריצה הוא: $O(|E|\log|V|)$.

אם דקסטרה עובר על גרף כמו BFS, מדוע הסיבוכיות שלו גדולה יותר ?

איך אפשר לאכסן גרף בתוך ערימת פיבונאצ'י ?

איך אפשר לייצג את דייקסטרה ע"י ספר מטריצות ?
מכפילים את מטריצת השכנויות של הגרף בעצמה מספר רב של פעמים
במקום + עושים כפל, ועוד מספר שינויים מופיע בקורמן

האם אפשר להתייחס לדייקסטרה כתכנון דינאמי ? ו DFS ?
כן
כן

איך מזהים שאלת דקסטרה ?
גרף לא מכוון עם משקלים אי שליליים

בגרף לא מכוון עם קשתות במשקלים אי שליליים, איך מחזירים קבוצת צמתים הנמצאת על מסלול אופטימאלי כלשהו משני צמתים נתונים ?
פתרון א :
ריצת דקסטרה רגילה
ריצת דקסטרה בכוון השני
לעבור על כל הצמתים ולבדוק אם סכום המחירים ממנו להתחלה וממנו לסוף שווה למחיר הזול הכולל מההתחלה לסוף (או מהסוף להתחלה)

מהו גשר ?
צומת שבלעדיו הגרף לא קשיר

איך אפשר לגרום לדקסטרה להתמודד עם אילוצים נוספים, כמו צבעים על קשתות ?
או להשתמש בדקסטרה כקופסא שחורה ולבצע עוד פעולות או טריקים, כגון רישום של גרף חדש המכיל את הגרף הצבעוני והחוקיות הנוספת, כך שאפשר להפעיל עליו דקסטרה
או "להכנס" לדקסטרה ולנתח לו שינויים מבפנים

מהי השיטה של דייקסטרה ?
בהתחלה מסמנים את כל הקודקודים כאילו לא ביקרנו בהם ומרחקם מוגדר כאינסוף.
כל עוד נותרו צמתים שלא ביקרנו בהם:
- נסמן את X בתור קודקוד שביקרנו בו.
- עבור כל צומת Y שהוא שכן של X וגם לא ביקרנו בו
- נעדכן את Y כך שמרחקו יהיה שווה לערך המינימלי בין:
- מרחקו הנוכחי ומרחק X ממנו בתוספת המרחק בין בין S (צומת המקור) ל-X.
- בוחרים קודקוד X חדש – הקודקוד שמרחקו בשלב הזה הוא הקצר ביותר מבין כל הקודקודים בגרף שטרם ביקרנו בהם.

האם : המסלול הזול ביותר מ-S ל-T מכיל בתוכו את המסלולים הזולים ביותר בין כל שני צמתים ששותפים למסלול בין S ל-T ?
כן
הוכחה: אם זה לא היה ככה, היינו מחליפים את אותו קטע במסלול זול יותר, וזה בסתירה לכך שהמסלול הנתון הוא הזול ביותר.

מה זה "אם פי קומפליט " ?
סיבוכיות אקספוננציאלית ב k ?

למה, עבור אותה השאלה, לא פשוט :
+ נבצע מעבר BFS על הגרף.
+ עבור כל צומת נשמור את משקל המסלול עד אליה
+ במידה ונגיע לצומת בפעם השנייה – נשווה את המשקל הקיים למשקל החדש, ונשמור את המינימום.
?

האם דייקסטרה שומר את הצומת האחרון במסלול לצומת הנוכחי ?
לא ?

כיצד ממומש דייקסטרה ע"י ערימת מינימום ?

למה בעצם צריך את ההשוואה בדייקסטרה, הרי בטוח שהפעם הראשונה שבה אנו מגיעים לצומת מייצגת את המסלול הקרוב ביותר ?

האם : אם משקל כל הקשתות יגדל ב k, אז משקל המסלול הקל ביותר יגדל בכפולה של k ?
לא
ייתכן שמסלול קצר יותר בקשתות יהיה מסלול הכי קל חדש

האם הוספת ערך קבוע לכל הקשתות עלול לשנות את המסלול הזול ביותר ? למה ?
כן, כי תלוי באורכו.

Bellman Ford

מהו אלגוריתם בלמן פורד ?
מוצא מסלול זול עבור גרף עם משקולות
הכוללים שליליים, כאשר נתון אורך מירבי
למסלול

מהו רצף הטענות לאלגוריתם ?
+ כל המרחקים "מוגדרים" (גדולים
מאינסוף) או"א אין מעגלים שליליים
+ אם כל המסלולים מוגדרים אז המסלול
הזול הוא מסלול פשוט ובו לכל היותר n-1
קשתות
+ אם אין מעגלים שליליים מתקיים
d(s,u) = d(n-1) (s,v)
+
?

מהי שיטת האלגוריתם ?
דומה לדייקסטרה
אבל אם מגלה שהגיע מאותו כוון ומקבל
מסלול זול יותר, עוצר ומכריז על מעגל שלילי
?
עבור כל צומת מבצע דייקסטרה פרטי, עבור
מסלולים באורכים שונים ?

מהי סיבוכיות האלגוריתם ? למה ?
O(|V| *|E|)

מהם הסימנים שזהו האלגוריתם שבו עלינו
להשתמש ?
מעגלים
משקולות שליליים

האם בלמן פורד מוצא מעגלים ? איך ?
כן
?

ממה נובע ההבדל בסיבוכיות בין בלמן פורד
לדייקסטרה ?

מסלולים קלים בגרף מכוון ללא מעגלים

כיצד עובד האלגוריתם ?
מיון טופולוגי, איתחול הגרף לאינסוף, ואז מעבר על הקודקודים לפי הסדר הטופולוגי שלהם, ובמהלכו, הקלה על כל הקשתות השכנות.

מהי יעליות האלגוריתם המחשב מסלולים קצרים בגרף מכוון ללא מעגלים?
לינארית לפי גודל הגרף

מקבים ?
מסלולים קצרים ביותר

DFS – חיפוש לעומק

DFS ? כיצד עובד האלגוריתם ?
depth first search
מחסנית, זמן רץ,

מהם השימושים ל DFS ?

מה ההבדל בין BFS ל DFS ?
חיפוש לרוחב לעומת לעומק
סיבוכיות

תכונות DFS ? הוכחות לתכונות ?
מגלה את כל הצמתים שניתן להגיע אליהם
מ s – הוכחה באינדוקציה
הקשתות ה
מסומנות מהוות עץ מכוון ששורשו s
לכל צומת v שנתגלה תת העץ של v מכיל
בדיוק את כל הצמתים שניתן להגיע אליהם
מ v ועדיין לא נתגלו
תכונת הסוגריים מחסנית
אם x התחיל אחרי v אז x יהיה בתת העץ
של v

מהי תכונת הסוגריים, תכונת המחסנית ?
צומת שהתגלה אחרי צומת נתון, יצא לפניו.
אין שני צמתים שחייהם נחצים

האם DFS דטרמיניסטי ?
בהנחה שסדר הקשתות מצומת קבוע, אז כן

מהן גרסאות ל DFS ?
רקורסיבית
אלג' טרמקס – סימון מעברים
הופקרפט וטרזן – סימון צמתים, קשתות,
אב, בן,

d(v) ?
זמן כניסה ויציאה

כמה פעמים נכנסת כל צומת למחסנית ב
DFS ? BFS ? למה ?
1

מה הרעיון של ה DFS ? למה ?
לייצר עץ מכוון המגיע לכל הצמתים
עדות לאפשרות הגעה

האם DFS מגלה מסלולים קצרים ? BFS ?
לא
כן

איך בוחרים צומת התחלה ל DFS ?

האם ב BFS מקבלים את עץ המסלולים
הקצרים ?
בדרך כן אבל לא בהכרח

מהם סוגי הקשתות שיוצר בעצם DFS ?
קשתות העץ : גילוי ראשון של צומת
קשתות אחוריות : מצאצא לאב קדמון
קשת קידמית : קשת מקצרת, מאב קדמון
לצאצא, תוך דילוג על דור אחד לפחות.
קשתות אלו אינן קשתות עץ.
קשת חוצה : קטעים חיים זרים

האם לקשתות יש זמן ?
לא

אם קיימת קשת חוצה, מה אפשר להגיד על
זמני עץ המקור והיעד ?
עץ היעד היה לפני עץ המקור

האם DFS עובד עבור גרף לא מכוון ?
כן אך עם מעט הבדלים
יהיו קשתות עץ
קשתות אחוריות וקדמיות יהפכו לאותו
הדבר, ויקראו קשתות אחוריות
לא יהיו קשתות חוצות כי אין להן כוון
שמאפשר זאת. באלגוריתם יימצא עץ אחד
כתת העץ של השני

מהי סיבוכיות DFS ? למה ?
$O(|V| + |E|)$
צביעת כל הקודקודים בגרף בלבן מתבצעת
בזמן $O(V)$, בדיקת כל קודקוד מתבצעת
בידיוק פעם אחת לכל קודקוד שכן הבדיקה
מבוצעת רק על קודקודים לבנים, והדבר
הראשון בבדיקת קודקוד היא צביעתו
באפור.
במהלך בדיקה על קודקוד נריץ שוב את
הבדיקה על כל שכניו, מאחר וסכום כל
השכנים של כל הקודקודים הוא $O(E)$ העלות
הכוללת של כל בדיקות הקודקודים היא
$O(E)$.
זמן הריצה הכולל של חיפוש לעומק הוא
$O(E+V)$

?,,,,,,,

2 DFS

מהם שימושי DFS ?
מציאת מעגלים
מציאת קשירות בגרף
בעיית ההחלטה

מה עדיף, DFS או BFS ?
בעקרון מחסנית יותר נוחה ממערך, ויותר יעילה מבחינת זכרון

מהי בעית ההחלטה ?
האם הגרף מכיל מעגל מכוון ?

מה מאפיין DFS ללא קשתות אחוריות ?

איך מריצים DFS על גרף לא קשיר ?
מריצים שוב על צומת לא קשיר לריצה הראשונה, ללא איפוס הזמן

מהו DFS מוכלל ?
עבור גרף לא קשיר
מריצים DFS מצומת שרירותית.
אם בסיום יש צמתים שלא סומנו, בוחרים אחת, מכניסים למחסנית, וממשיכים ריצת DFS ללא איפוס הזמן

מה סיבוכיות של DFS מוכלל ?
כמו DFS רגיל

מהו יער עומק ?
שלא כמו בחיפוש-לרוחב, תת-גרף הקודמים הנוצר ע"י חיפוש-לעומק עשוי להיות מורכב מכמה עצים ולא עץ אחד בלבד ולכן נקרא לו יער-עומק המורכב ממספר עצי-עומק.
(הדבר עשוי לקרות מכיוון שחיפוש-לעומק יכול להתנהל ממספר מקורות).
?

מהי חלוקת יער העומק של הגרף ?
לפי קשתות העץ ?

מהם "מעברים" ?
חיבורים בין קשת לצומת

האם ב DFS מסמנים צמתים או מעברים ?
למה ?
מעברים, אחרת לא נוכל לדעת על קשתות שלא ביקרנו בהן ?

מה הקשר בין מימושי DFS ו BFS למחסנית ותור ?
DFS מבצע בעצם מחסנית LIFO
BFS מבצע בעצם תור FIFO
?

האם BFS ו DFS מייצרים את אותו עץ פורש ?
לא כי DFS יגיע קודם לצמתים שבמסלול הראשון, ו BFS יגיע אליהם במסלול הקרוב ?

מהו k(v) לפי טרזן ?
אינדקס סדר הגילוי של צומת

האם DFS הוא דטרמיניסטי ?
כן אם מקפידים על חוקיות בסדר, אבל לא עבור גרפים שקולים ?

DAG

איך נקרא גרף מכוון חסר מעגלים מכוונים ?
DAG
directed acyclic graph

מהם שימושים של DAG ?

מה המשמעות של מעגל גרירה ?
שקילות לוגית בין כל הטענות הלוגיות

האם : גרף מכוון הוא DAG או"א ב DFS
מוכלל אין קשת אחורית ? למה ?
כן !
צומת כלשהו במעגל חייב היה להתגלות
ראשון. כל שאר הצמתים שניתן להגיע
אליהם ממנו נמצאים בתת העץ שלו (תכונת
DFS). לכן גם הצומת מהקשת האחורית שלו
נמצאת בתת העץ

אם מורידים קשת מ DAG, האם הוא נשאר
DAG ? וצומת ?
כן
כן ?

מהו האלגוריתם השני למציאת DAG ? מה
הסיבוכיות שלו ?
?
לינארית

האם : כל יחס סדר סופי מגדיר DAG וכל
DAG מגדיר סדר ? למה ? מה המשמעות ?
כן
?

איך מוציאים צומת ממבנה נתונים של גרף
בסיבוכיות נמוכה ? מהי ?
בזמן בניית מערך הרשימות המקושרות -

מיון טופולוגי

מהו מיון טופולוגי ?
המיון הוא סידור לינארי של קודקודי הגרף
כך שאם קיימת קשת מקודקוד אחד לשכנו,
הקודקוד יופיע לפני שכנו במיון הטופולוגי.
ניתן לראות מיון טופולוגי של גרף כסידור של
קודקודיו לאורך קו אופקי דימיוני, כך שכל
הקשתות המכוונות פונות משמאל לימין.
מעין חלוקת גרף לרמות, המיון מוסיף לכל
צומת נתון נוסף, מעין "דרגה" המייצגת את
מיקום הצומת במיון.
?

מהו עקרון הבסיס של מיון טופולוגי ?
אם קיים מסלול מ u ל v אז u נמצא לפני v
במיון

מהו האלגוריתם למיון טופולוגי ?
הרצת DFS והחזרת זמן הסיום עבור כל
צומת, ומיון הצמתים עפ נתון זה

מהם התנאים לגרף הנתון עבור מיון
טופולוגי ?
גרף מכוון ללא מעגלים בלבד

על איזו סריקה מסתמך המיון הטופולוגי ?
סריקה לעומק DFS

מה ההבדל בין מיון טופולוגי למעודן ?

מה ההבדל בין מיון טופולוגי של אילן למיון
טופולוגי של הספר ?

מהו האלגוריתם של מיון טופולוגי ?

מהי סיבוכיות האלגוריתם מיון טפולוגי?
כמו DFS
O(V+E)

האם מיון טופולוגי עובד רק על משקלים
שלמים על הקשתות ?

מה ההבדל בין מיון טופולוגי למעודן ?
?? - לכל צומת שכבה משלו
מעודן – ייתכנו מספר צמתים באותה שכבה

איך מחושב מספר המסלולים מ x ל y ב
DAG ?
תכנון דינאמי מהסוף להתחלה על המיון
הטופולוגי
כל צומת הוא סכום הצמתים שאחריו

רכיבי קשירות

מהו רכיב קשירות בכלליות ?
מבטא מסלול הקיים בין אזורים בגרף

רק"ח ? מה משמעותו ?
רכיבי קשירות חזקה (\היטב)
קבוצת קודקודים מקסימלית בתוך הגרף, בה יש מסלול מכל קודקוד לכל קודקוד

למה חשובים רכיבי קשירות ?

נגדיר יחס
גשר יחיד בין שני גרפים קשירים חזק ?

אם אפשר להגיע בגרף מכוון

מה הקשר ליחס שקילות ?

מהן מחלקות השקילות ?
המתקבלות מהיחס הבא :
האם יש מסלול מ u ל v ולהיפך ?

איך בדיוק מבוצעת החלוקה למחלקות השקילות ?

מהי מחלקת שקילות ?
טרנזיטיבי, רפלקסיבי ואנטי סימטרי ?

מהו גרף הופכי ?
גרף המכיל את אותם הקודקודים, אך כל קשתותיו פונות לכיוון ההפוך

האם : רכיבי הקשירות של גרף הופכי הם בדיוק אותם רכיבי הקשירות של הגרף המקורי ? למה ?
כן
עדיין ניתן להגיע מכל צומת אל עצמו דרך כל שאר הצמתים בקבוצת הקשירות, רק על המסלול ההפוך

יחידת קשירות ?

מהו גרף קשיר היטב ?

מה בעצם מייצגות מחלקות השקילות ?

מה הקשר בין קשירות ל DAG ?

מהו זמן ריצת האלגוריתם למציאת רכיבי קשירות?

DFS
מציאת גרף הופכי
DFS על הופכי
סהכ מספר קבוע של פעמים זמן דומה
O(V+E)

מהו אלגוריתם מציאת רכיבי קשירות ?
+ DFS
+ מציאת גרף הופכי
+ DFS לגרף ההופכי, עם שינוי קטן
מהאלגוריתם המוכר : בזמן בחירת הקודקודים, נבחר לפי סדר הסיום היורד שלהם שהתגלה ב DFS על גרף המקור.
כל עץ ביער העומק אחרי ה DFS על הגרף ההופכי הוא למעשה רכיב קשירות חזקה.
כל עץ ביער החדש הוא רכיב קשירות

איך מזהים עץ ביער העומק אחרי ה DFS השני ?

שידוך

מהו שידוך ?
זוג קשתות שאין להן קצב משותף

למה שידוך הוא עניין חשוב או מעניין ? אילו שימושים יש לו ?

מה ההבדל בין גרף מכוון לגרף דו צדדי ?

מהו אלגוריתם התכנון הדינאמי למציאת שידוך מרבי בעץ לא מכוון ?

מהי סיבוכיות המקום של אלגוריתם התכנון הדינאמי של מציאת שידוך מירבי ? האם לא נכון שיש צורך בהקצאת מקום רקורסיבית ולכן גדולה יותר ?

מהו הטוויסט של אלגוריתם התכנון הדינאמי שהופך את הסיבוכיות מקשתות לצמתים ?
עבודה על צמתים במקום קשתות
פיצול לשני מקרים – שידוך מכיל את הצומת ושידוך שלא מכיל

איך מתמודד האלגוריתם עם שווין במשקל בין תת עץ מכיל ולא מכיל ?
האלגוריתם נותן תשובה מספרית ולא מפרט באילו קשתות בחרנו
אילו היינו רוצים לדעת באילו קשתות מדובר היה עלינו לבחור אקראית במקרה שוויון ולשמור את הקשתות שבחרנו לאורך האלגוריתם.
איך מוצאים מהו מספר השידוכים הגדולים ביותר ?
נוסיף לאלגוריתם קאונטר
בכל מקום שבו היה שוויון נכפיל את הקאונטר ב 2

מהי סיבוכיות האלגוריתם המוצא את מספר השידוכים המכילים k קשתות עבור k בין 1 ל n/2 ? למה ?
ריבועית
n^2 לכל קשת ובסה"כ n^3

מהו שידוך מושלם ? מהו מושלם לפי קבוצת הבנות ?
מכיל חצי מהקשתות בגרף
גודל השידוך כגודל X

מהו האלגוריתם לשידוך מיטבי בגרף מכוון דו צדדי ?
+ מוסיפים קודקוד חדש שמתחבר לכל הצמתים בצד אחד, וקודקוד דומה לצד השני. כוון הקשתות – מהקודקוד החדש אל הצמתים הקיימים. קיבולות – 1 על כל קשת.

+ בקיבולות G מציבים אינסוף

האם : אם יש שידוך M בגודל k ב G, אז יש זרימת s-t בגודל k ב G' ? למה ?
כן

האם : אם יש זרימה בגודל k ב G' אז יש שידוך בגודל k ב G ? למה ?
כן

מה הקשר בין זרימה לשידוך ?

האם : אם יש f בגודל k ב G' אז אפשר למצוא k מסלולים זרים בין s ל t ? למה ?
כן
מסלולים אלו גם זרים בצמתים למעט s.t,
ולכן הם מגדירים שידוך בגודל k

מדוע k מסלולים זרים מגדירים שידוך בגודל k ?

האם : אלג' למציאת זרימת מקס' מ s ל t מוצא שידוך מירבי ב G' , וגם את השידוך עצמו ? למה ?

מהו משפט Hall ?
לגרף דו"צ G = (X,Y ; E) יש שידוך מושלם המשדך את X (הבנות) או"א
for every U belog to X, |N(U)|>=|U| when
N(U)={v blg Y |there is u blg X, (u,v)blgE}

מה עקרון ההוכחה למשפט הול ?
כוון אחד טריוויאלי
כוון שני הנחה בשלילה

איך יודעים מהו השידוך לפי הזרימה ?

מהן הקיבולות שנותנים לקשתות ב G ?
אינסוף

מה עוד נובע ממשפט הול ?
אפשר למצוא את גודל השידוך גם כשאינו מושלם ?

כמה חלוקות ישנן מגרף נתון לגרף דו צדדי שנובע ממנו ?

מהו חיתוך בגרף ?
קבוצת הקשתות שביטולן ינתק מהגרף
קבוצת צמתים נתונה

מהו חתך מינימום ?
קבוצת הקשתות הנתקת המינימלית

גודל הזרימה ?

זרימת מקס בחתך מינ' שווה לערך החתך ?
כן

מהי קבוצת כיסוי קשתות ?
תת קבוצה של צמתים שפוגשת את כל הקשתות

מהי בעיית הכיסוי המינימלי vertex cover ?
מציאת קבוצת כיסוי מינימלית

מהו הפתרון לבעיית הכיסוי המינימלי ?
א – הכיסוי המינימלי מוגבל מלמטה ע"י גודל השידוך המקסימלי

מהו משפט קניג ? למה ?
בגרף דו צדדי גודל שידוך המקס' שווה לגודל כיסוי המינימום
+ אם השידוך מושלם, קבוצת הבנות היא קבוצת כיסוי
+ אם |X| < |M| אז (X-A)U(YcutA) קב' כיסוי בגודל זה

מהו האלגוריתם למציאת מסלול אוילר משתנה בקשתות ?
אלגוריתם רגיל למציאת מסלול אוילר עם התנאי – בחירת הקשת הבאה לפי הצבע שנשארו ממנו הכי הרבה קשתות
?

עץ פורש מינימלי M S T

מהו עץ פורש ?
גרף קשיר ללא מעגלים המכיל את כל הצמתים בגרף נתון קשיר

MST ?
minus spanning tree

מדוע לא כדאי לקחת את עץ המסלולים הזולים ביותר ? דוגמא ?
כי זה לא נכון
זה תלוי צומת התחלתית

האם בהכרח, עץ פורש מינימום מכיל את המסלולים הקצרים בין כל שתי קשתות בגרף המקורי ? למה ?
לא ?
כי יכול להיות שהורדנו קשת שסוגרת מעגל שהיקפו קטן מהקשת שהורדנו

כמה אלגוריתמים ישנם למציאת עץ פורש מינימום ? האם הם חמדניים ?
5 עיקריים ?
כן ?
קרוסקל
פרים

כמה קשתות בעץ פורש ?
n-1

מהו מספר המסלולים בין כל שני צמתים בעץ פורש ?
1

מהו תת גרף מקסימלי ? מה הקשר ?
?
עץ פורש הוא תת גרף מקסימלי של G שהינו חסר מעגלים

כמה מעגלים נוספים בהוספת קשת בעץ פורש ? למה ?
1

מהו C(T, e) ?
מעגל הנוצר מעץ פורש T וקשת e שהוספה לו

האם : בכל עץ פורש מינימלי הקשת שהוצאנו היא הגדולה ביותר ? למה ?
כן
אחרת היינו מוציאים קשת אחרת

האם : עץ הוא MST או"א כל קשת שנוסיף גדולה מכל הקשתות במעגל אליו היא שייכת ? למה ?
כן

חמדני ?

איך בוחרים צומת התחלה לפריסת עץ ?
+ אם בונים את העץ הפורש ע"י הורדת קשתות אז אין צורך לבחור צומת התחלה
+ אם האלגוריתם ליצירת עץ מינימלי הוא נכון, כמו פרים, אז לא משנה מה תהיה צומת ההתחלה, תמיד יתקבל אותו העץ

כמה עצים פורשים ייתכנו עבור גרף ?
הרבה ?

האם לכל גרף קיים עץ פורש ללא פיצולים ?
לא ?

בהנתן גרף לא מכוון, ממושקל אשר קשתותיו צבועות בכחול או בלבן.
כיצד, ע"י שינוי קטן מאוד באלגוריתמי מציאת עפמי"ם, תוכל למצוא עפ"מ בעל מספר קשתות כחולות מקסימלי ?
בזמן בחירת הקשתות, קשת כחולה תבחר לפני לבנה בעלת אותו משקל

האם : עבור כל עץ פורש, הוספת קשת חדשה כלשהי בהכרח תיצור מעגל ? למה ?
כן

האם : עבור כל עץ פורש שהוספנו לו קשת, הוצאת קשת אחרת מאותו המעגל תיצור עץ פורש חדש ? למה ?
כן

האם : אם הקשת שהוספנו היתה שייכת לגרף הנתון עבורו מצאנו עץ פורש מינימלי, אזי הקשת גדולה משאר הקשתות במעגל ?
כן

מהו חתך בגרף ?
קבוצת הקשתות שמחיקתן מ G מנתקת כל צומת ב S מכל צומת שאיננה ב S

האם : חיתוך C(s) ו – C הוא נותן קבוצה שעצמתה זוגית ? למה ?
כן
?

האם : T הוא mst של G או"א הקשתות הגדולות בכל מעגל אינן בעץ הפורש ? למה ?
?
כן
עקרון ההוכחה : דו כווני. א' טריוויאלי, ב' ע"י הנחה בשלילה

מהו אלגוריתם הכווץ ל mst ?
+ בחירת קשת קלה ביותר בגרף
+ איחוד צמתי הקצה לצומת אחד
+ מחיקת לולאות עצמיות
+ חזרה על הפעולה עד שנשארים עם צומת אחת בגרף

האם : אם כל המשקולות בגרף שונים אזי קיים mst יחיד ? למה ?
כן

מהו ה mst עבור אוסף משקולות שונים ?
v-1 המשקולות הטובות

כמה mst קיימים עבור גרף נתון ?
אחד ומעלה

נוספה לגרף G צלע. איך נעדכן את ה mst ?
בכמה זמן ?
נשווה לשאר הקשתות במעגל שיצרה ונוציא את הגדולה
O(|E|) ?

פרים וקרוסקל

מהו האלג' של קרוסקל ?
הוספת קשתות מהקלות לכבדות כל עוד
אינן יוצרות מעגל
עד שמוסיפים v-1 קשתות

מהו האלגוריתם של פרים ?
מתחיל מצומת שרירותית
מוסיף כל פעם את הקשת הקלה המחוברת
לאחד הצמתים שכבר חוברו
?
דומה לדייקסטרה

מהי הסיבוכיות של קרוקסל ? למה ?
הכנה, מיון הקשתות ב O (E logE)
איחוד הקשתות במבנה נתונים union find
מאפשר זמן כולל של O(|E|log n)
?
ויקיפדיה – O(|E|log|E|) למיון הקשתות

מהי הסיבוכיות של פרים ? למה ?
כמו של דייקסטרה
O(|V|log|V|+|E|)
במימוש ע"י ערימת פיבונאצ'י

מהם התנאים להפעלת קרוקסל ? פרים ?
גרף לא מכוון וממושקל ?

מהו תנאי העצירה עבור קרוקסל ? פרים ?
V-1 קשתות

האם יעילות האלגוריתם של קרוסקל
תשתפר אם נדע מראש כי משקלי הקשתות
יכולים להיות רק 1 ו-2 ? כמה למה ?
כן
מיון הקשתות יקח O(E)
סכה יהיה O(V+E)

מהו פירוט האלגוריתם של קרוקסל ?
2-3: עבור כל קשת בגרף ניצור קבוצה
(המכילה איבר אחד – את הקשת עצמה).
4: מיון של הקשתות לפי משקל
5: עבור כל קשת (u,v),החל מהקשת הקלה
ביותר ועד הכבדה
6: אם u ו-v לא שייכים לאותה קבוצה
7: נוסיף ל-A את (u,v)
8: נאחד את הקבוצות של u ושל v לקבוצה
אחת.

למי מהאלגוריתמים סיבוכיות טובה יותר ?
מה החריגה ?
בעקרון של פרים טובה יותר
אלא אם הקשתות כבר ממוינות לפי גודלן
ואז לפי של קרוסקל

מהו זמן ריצה

? O(E alfa (E, V))

האם בהכרח, אם נבחר v-1 קשתות ללא
מעגלים מגרף, נקבל עץ פורש וקשיר של
הגרף ?
כן ?

רשתות זרימה

אילו שימושים או מודלים יכול הנושא להוות ?
צינורות ומחברי מים
קוי אספקת חשמל
גז
ועוד

מהי רשת זרימה במושגי גרפים ?
גרף מכוון שבו כל קשת היא בעלת קיבול אי שלילי (נסמן את הקיבול ב c), ברשת הזרימה קיימים שני קודקודים מיוחדים: מקור s ובור t. אנו מניחים שכל קודקוד שוכן על מסלול כלשהו מהמקור לבור.

מהו משקל על הקשתות ברשתות זרימה ?
ישנם שני נתונים, קיבול מירבי (פוטנציאל, אילוץ קיבול) וזרימה בפועל

מדוע זרימה היא פונקציה ?

מהן תכונות הזרימה ?
1. אילוץ קיבול: הזרימה על קשת אינה יכולה להיות גדולה מהקיבול על קשת זו.
2. סימטריה נגדית - הזרימה על קשת שווה למינוס הזרימה על אותה הקשת בכיוון ההפוך.
3. שימור הזרימה - סכום הזרימה על כל הקשתות הסמוכות לקודקוד מסויים שווה ל-0.

מהי משמעות תכונה 2 אם אין זרימה שלילית ?

מהו ערך הזרימה ?
ערך הזרימה של פונקצית זרימה הוא סך הזרימה היוצאת מקודקוד המקור.

כיצד מוצאים את ערך הזרימה המקסימלי ? אפשר מהמקור לבור או להיפך ?

מהי בעיית הזרימה המקסימלית הכללית ?
מספר מקורות ומספר בורות

מהי בעיית חתך המינימום ?

האם max_flow <= min_cut ? למה ? מה המשמעות ?
כן
?

מהו C(s) ?
קבוצת הקשתות החותכות מהאזור הקשיר של s לאזור הקשיר של t ?

מהי זרימת מקסימום ?

מהו חתך מינימום ?
קב' הקודקודים שעד אליהם יש מסילה מגדילה

האם מסילה מגדילה בוחרים עבור קשת או עבור מסלול מלא ?
לפי קשת ?

c ?
קיבולת

מהו התנאי לזרימת מכסימום חתך מינימום ? למה ?
$f(s) = w(C(s))$
?

האם : f זרימת מכסימום או"א אין ביחס ל f מסילה מגדילה ? למה ?
כן
כי לא ניתן להזרים יותר חומר, כי אין איפה
?

האם : max flow = min cut ? למה ? מה המשמעות ?

מהי זרימה חוסמת ?

האם : אם f שלמה (כל הערכים שלמים) וכל הקיבולות שלמות, אזי יש מסילה מגדילה ביחס ל f, ניתן לשפר את הזרימה ולהגיע לזרימה חדשה שלמה גם כן ? למה ?
כן

האם : ברשת בה כל הקיבולות שלמות יש זרימת מכסימום שלמה ? למה ?
כן

האם ייתכן מצב שבו, בזרימה מקסימלית, קשת רוויה אינה קריטית להקטנה ? למה ?
כן !
למשל אם צוואר הבקבוק הוא בקשת האחרונה ויש יותר ממסלול אחד עד אליה

האם ייתכן "מסלול שיפור טועה", מסלול שהפעלתו יכולה לגרום לפגיעה בזרימה המקסימלית האפשרית בגלל פגיעה במסלולי שיפור עתידיים ?
לא ?

איך מוצאים חתך מינימלי ?
מציירים את הרשת השיורית.
מוצאים חזית bfs ת של אפסים.

האם כל קשת רויה שייכת לחתך מינימום כלשהו ?
לא

האם בהכרח קיימת כל זרימה קטנה מהזרימה המרבית ? למה ?
כן
מוסיפים עוד צומת אחרי הבור וקשת עם הקיבולת הקטנה יותר, ומוצאים זרימה מרבית

האם זרימה בינארית מרבית מתארת מספר מסלולים מרבי ? למה ?
לא
?

האם כל קשת קריטית להקטנה היא גם קריטית להגדלה ? ולהיפך ?
לא
לא

האם בכל רשת קיימת קשת קריטית הקטנה ? למה ?
כן
כי קיים חתך. אלו אותן קשתות

האם תתכן קשת שהזרימה עליה גדולה מהזרימה המרבית ? למה ?
לא
כי מהמקור לא יוצאת יותר מהזרימה המרבית, אז לא ניתן ליצור כזו זרימה ?

האם כל קשת קריטית להקטנה היא חלק מחתך כלשהו ? וקריטית להגדלה ? למה ?
כן
לא
?

שיטת פורד-פולקרסון

מה עקרון האלגוריתם ?

מהי סיבוכיות האלגוריתם ?
עבור זרימה עגולה
$|E|*f$
f היא הזרימה המרבית

מה משמעות המילה שיורי ?
נותר ?

מהו גרף שיורי ?
גרף שיורי הוא בראש ובראשונה היפוך כל הקשתות שבהן היתה זרימה בגרף המקורי. כאשר משקל הקשתות יהיה כזרימה שהייתה עליהן, והקשתות המקוריות של הגרף יקבלו זרימה 0 וקיבול כקיבול המקורי פחות הזרימה שהייתה.
בהינתן רשת זרימה וזרימה, הרשת השיורית מורכבת מקשתות שיכולות להכיל זרימה נוספת. הכמות הנוספת של זרימה שניתן להוסיף בין שני הקודקודים היא הקיבול השיורי.
הזרימה המועברת דרך כל קשת ברשת השיורית היא חיובית ממש.

מהו קיבול שיורי ?
הקיבול בין שני הקודקודים פחות הזרימה הנתונה בין שני הקודקודים.

מהי שיטת פורד פולקרסון ?
אלגוריתם איטרטיבי למציאת זרימה מקסימלית בעזרת גרף שיורי
שיטה מקיפה זו מבוססת על שלושה רעיונות חשובים הממלאים תפקיד מכריע במשפט זרימה-מקסימלית-חתך-מינימלי

מהו מסלול שיפור ?
בהינתן רשת זרימה וזרימה, מסלול שיפור הוא מסלול פשוט מהמקור אל הבור ברשת השיורית.

כיצד עובדת השיטה ?

כיצד מיוצר גרף שיורי מגרף נתון לפי השיטה ?
+ הוספת קשת הפוכה לכל הקשתות שהיו חסרות משקל
+
+ מחיקת קשתות במשקל אפס
??

כיצד נקבע המשקל עבור קשתות חדשות בגרף שיורי ?

האם ישנו שינוי כלשהו בקשתות הגרף המקורי ביחס לשיורי ?
לא

מהו אלגוריתם איטרטיבי ?
אלגוריתם שבו פונקציה מוצאת פתרון לבעיה ע"י הרצת אותה פונקציה שוב ושוב (איטרציות) על חלקים קטנים יותר של הבעיה, עד למציאת הפתרון.

מהו משפט זרימה-מקסימלית-חתך-מינימלי ?
משפט חשוב אשר מאפיין את ערכה של הזרימה המקסימלית במונחי חתכים של רשת הזרימה.
משפט הזרימה-המקסימלית-חתך-מינימלי מראה כי הזרימה המתקבלת בסוף האלגוריתם אכן מקסימלית
?

מהו אלגוריתם השיטה ?
+ מתחילים עם זרימה התחלתית שערכה אפס, או זרימה חוקית אחרת כלשהי.
+ בכל איטרציה מגדילים את ערך הזרימה ע"י מציאת "מסלול שיפור" - מסלול מן המקור לבור שלאורכו ניתן להגדיל את הזרימה. למשל ע"י BFS.
+ חוזרים על תהליך השיפור עד שלא ניתן למצוא מסלולי שיפור.
+ עם סיום התהליך היא הזרימה המקסימלית.

איך "מוסיפים זרימה" על קשת ?
מוסיפים לערך הזורם, בתנאי שלא עובר את האילוץ

את "מקזזים לזרימה אחת" ?
מורידים את הערך הזורם בכוון שזורם פחות ל 0, ומקזזים ערך שווה בכוון השני

מה עושים אם צריך להזרים יותר מהאילוץ ?
אם אפשר – מפחיתים את ההפרש מהכוון השני

מהי ההנחה עבור השיטה ? מה המשמעות ?
קיבולות שלמות

מהי סיבוכיות האלגוריתם ? האם פולינומיאלית ? למה ?
$O(|V|+|E|) * C$
לא
כאשר C הוא סכום הקיבולות ששכנות ל S
?
שזה בערך $O(|V|^3)$?

האם סיבוכיות האלג' טובה ?
היא אינה פולינומיאלית, משום שיש לנו את סכום הקיבולות ולא ב log שלהם

האם : האלגוריתם של FF נכון למשקולות שלמים ? רק לשלמים ? למה ?
כן
לא ?

האם : כל זרימת מכסימום היא שלימה ? למה ?
לא

האם : אם המשקולות שלמים, זרימת המכסימום גם כן שלמה ? למה ?
כן
?

מה הבעיה עם האלג' ? מה הפתרון ?
+ יכול לקחת אינסוף צעדים, לא תמיד עוצר
+ ולא להתכנס תמיד לדבר הנכון
_ בתנאים של קיבולות שלמות וזרימה התחלתית שלמה האלג של FF עוצר תמיד ונותן זרימת מקס'

מהי דוגמת ה M ?
מדגימה את הסיבוכיות

מהו עקרון דו הכווניות של הקשתות המכוונות באלגוריתם ?
אפשר לזרום דו כוונית בתנאי שלא שלילי בכוון המוגדר

מהי הדרך המדויקת למציאת מסלול שיפור ?

BFS על הרשת השיורית, הרשת המכילה
את ההפרשים בין הקיבולת המירבית
לזרימה

אלג' אדמונדס קרפ

מהו האלגוריתם ?
כמו FF למעט בחירת המסלולים.
בשלב הראשון מבוצע bfs ומזרימים עליו לפי צוואר הבקבוק
באיטרציה הבאה נמחקת הקשת הרוויה ומבוצע bfs נוסף על הגרף השיורי החדש
בכל איטרציה נבחר מסלול חדש, ארוך יותר מקודמו
מספר הקשתות במסלול השיפור הנבחר קטן ככל האפשר, בכל שלב
?

מהי סיבוכיות האלגוריתם ? למה ?
O(|V| * |E|^2)
בכל הפעלה של מסילה מגדילה באלג' לפחות קשת אחת רוויה. בפעם השנייה שמשתמשים בקשת זו המרחק של אחד מקצותיה גדל בלפחות 2. לכן, קשת מסוימת יכולה להיות רוויה לאורך האלגוריתם לכל היותר O(|V|) פעמים.
יש סך הכל O(|V|*|E|) הגדלות.
בהנחה ש |V| =< |E|, כל הגדלה לוקחת O(|E|) .
בסה"כ נקבל O(|V| * |E|^2)
או O((|V|+|E|) * |V|*|E|)

על איזו בעיה בא האלגוריתם לענות ?
על הסיבוכיות הגבוהה של אלג FF
?

מה הכוונה ב "קשת רוויה" ?
שהזרימה בה שווה לקיבול, זרימה מירבית

איך ייתכן שלא מאבדים מסלולים בצורה זו
?

האם \ מדוע לא יותר הגיוני לבצע BFS למציאת המסלול הקצר ביותר ואז להאריך כל פעם ?

האם : lamda(fi)(s,v) >= lamda (fj)(s,v) ?
למה ?
כן
המרחק לא יורד מהפעלה להפעלה
צריך להסתכל על v הקרובה ביותר שלכאורה מתקרבת
?

למה בעצם האלגוריתם עובד ?
האלג' הוא מקרה פרטי של FF

למה הסיבוכיות קטנה בעצם ?
אפשר לחשוב על זה בתור מחיקת הקשתות ש"מתבטלות" בין השלבים
אבל בעצם נוספות קשתות עם זרימה הפוכה שמתאפשרת אחרי שיש זרימה כלשהי בכוון המוגדר

איך אפשר לדעת מהו מספר ההגדלות המירבי האפשרי ?
לכל צומת בכל ונגדיר – אורך הקצר ביותר של מסילת שיפור מ s ל v ב Gi ביחס ל fi

האם : לכל v ובפרט ל t סדרת המרחקים אינה יורדת ? מה המשמעות ? האם זה נובע מההגדרה ? למה ?
אורכי המסלולים גדלים או שווים לקודמיהם
לא
זה נובע מכך שיש קשת מתהפכת. יש איזשהו v שיורד
ההוכחה היא בשלילה, מניחים שהלמה לא נכונה. ז"א,
יהי i הקטן ביותר ועבורו v עם di(v) הקטן ביותר כך ש di(v) < di-1(v)
תהי x הצומת הלפני אחרונה במסילת השיפור הקצרה ביותר מ s ל v ב Gi
ולכן אם הקשת (x,v) belog Gi-1
di-1(v) <= di-1(x)+1 <= di(x) + 1 = di(v)
למה הקשת מ x ל v לא היתה ב Gi-1 ?
מסקנה 1 : מסלול השיפור ב Gi-1 עובר דרך הקשת (v, x)
מסקנה 2 : di-1(x) = di-1(v)+1
ולכן
di(v) < di-1(v) לפי ההנחה
di(v) = di(x) + 1 לפי האלגוריתם
di(x) >= di-1(x)
di-1(x) = di-1(v) + 1 מסקנה 2
וזו סתירה

האם בכל שלב בהכרח מופיעה קשת חדשה ? למה ?
כן, כי קשת אחרת הורוותה

האם : (v, x) הורוותה בשלב i, ואם בשלב j>i, משתמשים בה שוב (באיזשהו כוון) אזי d(v) >= di(v) + 2 ? למה ?

כן

האם : r, מספר ההגדלות באלגוריתם הוא לא יותר מ |E| * n פעמים.

סדר גודל סיבוכיות ראשונית ?
O(n^5)

האם ניתן להקטין ל O(n^3) ? למה ?
כן
סידור צמתים לפי מרחק מ s
צמתים שעברנו – המרחק אליהם גדל במסלול שיפור
שכבה היא צמתים באותו מרחק
להראות שמאורע טוב קורה באופן יותר תכוף

מהו השקר הקטן ?
איך מחליטים בכמה לשפר את מסילת השיפור
מספרים גדולים
?

אילו לא נמצא בשבוע
שעור השלמה ב 13

מה בעצם ההבדל בין האלגוריתמים ?
לאדמונד קרפ יש חסם – גודל הגרף
את פורד פלקרסון חוסם גודל הזרימה
כמו כן, FF מוגבל לזרימה עגולה

איך מוצאים חתך מינימלי ?
אם ידועה צורת הזרימה המרבית – אז עושים bfs על הגרף השיורי עד לחתך רווי מאופס

מתי עדיף FF על אדמונד קרפ ?
ייתכן מצב כזה כאשר נתונים אילוצים על הזרימות בקשתות

איך זרימה שקולה למטריצה ?
הוספת s ו t לגרף דו צדדי, כאשר X היא שורות ו Y עמודות

איך ייתכן שימוש חוזר בקשת רוויה באלג' EK ?
אם מבוצע שימוש בקשת הפוכה לרוויה זה בעצם מקטין על המקורית את הזרימה ?

האם ייתכן שמסלול חדש של EK לא ישפר
את הזרימה ?

קשירות וצביעת גרפים

האם : ברשת זרימה עם קיבולת 1 יש
זרימה בגודל k או"א יש k מסלולים זרים
בקשתות בין s ל t ? למה ?
כן

מה הכוונה במסלולים זרים ?
ללא קשתות משותפות ?

מהו משפט מנגר הצלע מכוון ? מה
משמעותו ?
בגרף מכוון יש k מסלולים זרים בקשתות בין
a ל b או"א אין k-1 צלעות המפרידות את a
מ b
?

מהו משפט מנגר ? מה משמעותו ?
בגרף (לא) מכוון יש k מסלולים זרים
(בקודקודים) בצלעות בין a ל b או"א אין k-1
(קודקודים) צלעות המפרידות את a מ b
?

מהו עקרון ההוכחה למשפט מנגר ?
רדוקציה לבעיה המכוונת צלעית
?

מהי צביעה חוקית של גרפים ?

האם : לגרף דו"צ d רגולרי קיימת צביעה ב
d צבעים ? למה ?
כן

מהו גרף רגולרי ?

האם : גרף בעל דרגה מכסימלית הגדולה
שווה ל d, ניתן לצבוע ב d+1 צבעים ? למה
? מה המשמעות ?
כן

שימושים לזרימה

כל הקיבולות הן 0 או 1, כמה זורם בכל
קשת ?
0 או 1

האם : ברשת 0\1 כל זרימה שלמה
מתפרקת לאוסף מסלולים זרים בקשתות מ
s ל t ועוד אוסף מעגלים (זרים בקשתות) ?
למה ?
כן

האם : עבור קיבולת K וזרימות שלמות
אפשר לצייר גרף דומה עם k קשתות כפולות
כמודל ? למה ?
כן

חתכים

מהו חתך ?
קבוצת צמתים קשירה הכוללת את s

מהו משפט מנגר ?
בגרף יש k מסלולים זרים או k-1 קשתות מפרידות.

מהו עקרון הוכחת משפט מנגר ?
max flow = min cut למכוון קשתות
ורדוקציות לכל שאר המצבים

מהי הכללת משפט מנגר ?
צמתים \ קשתות
לגרף מכוון \ לא

מהו הגרף המתאים לזרימה ?
מכוון

מהי רדוקציה ?
פתרון בעיה שאיננו יודעים לפתור בעזרת בעיה אחרת שאנו יודעים לפתור

איך הופכים גרף עם k מסלולים זרים לגרף שקול עם k קשתות זרות, ולהיפך ?
הפרדה וכיווץ של הצמתים והקודקודים המפרידים

האם : כמה צמתים מפרידים בגרף ובו k קשתות מפרידות ?

מהו סדר הרדוקציות להוכחת כל הוריאציות למשפט מנגר ? מה עקרון השחלוף ?
מכוון קשתות
מכוון קודקודים
לא מכוון צמתים – החלפת שתי קשתות מכוונות בקשת לא מכוונת, והפעלת רדוקציה לקודקודים
לא מכוון קשתות – החלפת קשת בגאדג'ט : הוספת שני צמתים, ובניית קשתות מכוונות שמאפשרות זרימה בשני הכוונים אבל לא בו זמנית

מהם השימושים במשפט מנגר ?
הוכחת השקילות על זוגות של קשתות שבינהן מעגל פשוט
סימטריות ורפלקסיביות נובעות מההגדרה
טרנזיטיביות : מחברים צמתים חדשים s ל u ו t ל h , כעת ניתן להוכיח בשלילה לפי משפט מנגר שלא קיים צומת מפריד בין s ל t ו f איפשהו באמצע הגרף החדש

גרפים ומטריצות

מהי ההמרה הטריוויאלית מגרף למטריצה ?
צמתים בשורות ועמודות
1 אם קיימת קשת ו 0 אם לא

מה משמעותן של חזקות למטריצה
הטריוויאלית ? למה ?
מספר מסלולים באורך החזקה מצומת
עמודה לצומת שורה !!
הוכחה באינדוקציה

נתון גרף לא מכוון וקשיר $G(V,E)$ עם משקלים על הקשתות, ונתון עץ פורש מינימלי של G. נניח שמוסיפים ל- G קודקוד חדש v וקשתות ממושקלות המחברות אותו לחלק מקודקודי הגרף. תארו אלגוריתם שרץ בזמן $O(|V|\log|V|)$ ומוצא עץ פורש מינימלי בגרף החדש.

__תאור האלגוריתם__

נריץ את האלגוריתם של קרוסקל למציאת עץ פורש מינימלי על הגרף שקבוצת צמתיו היא קבוצת כל הצמתים כולל הצומת הנוסף וקבוצת קשתותיו היא קשתות העץ המקורי ובנוסף כל הקשתות שמחברות את הצומת הנוסף עם הצמתים האחרים.

__הוכחת נכונות__

__טענה__: קיים עץ פורש מינימלי שבו אין אף קשת מהגרף המקורי שלא היתה בעץ הפורש המקורי.

נניח שבעץ פורש מינימלי כן קיימת קשת e_1 כזאת. נראה שניתן להחליף אותה על-ידי קשת שכן היתה חלק מהעץ המקורי, זאת בלי להגדיל את משקל העץ. אם נסיר את e_1 מהעץ אז יתקבלו שני רכיבי קשירות. בעץ המקורי קיימת לפחות קשת אחת שיכולה לחבר את שני רכיבי הקשירות האלה. נניח ש e_1 קלה יותר מכל קשתות העץ המקורי שיכלו לחבר את שני הרכיבים, אז יכולנו להוסיף את e_1 לעץ המקורי. בכך היה מתקבל מעגל שמכיל לפחות קשת אחת e_2 שמחברת את שתי הקבוצות. היינו מסירים את e_2 ומקבלים עץ קל יותר מהעץ המקורי. זאת סתירה. לכן בעץ המקורי קיימת לפחות קשת אחת שיכולה להחליף את e_1 בעץ החדש מבלי לגרום לעץ להיות כבד יותר.

__סיבוכיות__: בעץ המקורי היו $|V|-1$ קשתות.

כעת הוספנו עוד לכל היותר $|V|$ קשתות.

לכן זמן ריצת האלגוריתם היא $O(|V|\log|V|)$.

נתונה רשת זרימה $G(V,E)$ עם מקור s ובור t ובה זרימה f שערכה $|f|=1000$. האם בהכרח יש ברשת גם זרימה f' שערכה $|f'|=700$?

כן - נסתכל על רשת שקבוצת צמתיה היא קבוצת הצמתים של הרשת המקורית ובנוסף הצומת t_1 וקבוצת קשתותיה היא קבוצת הקשתות של הרשת המקורית ובנוסף קשת בעלת קיבול 700 מ t ל t_1. ברשת זאת נחפש זרימה מכסימלית מ s ל t_1. מכיון שבכל רשת הערך של זרימה מכסימלית שווה לערך חתך מינימלי, אז ברשת המקורית החתך המינימלי הוא של 1000. ברשת החדשה החתך המינימלי הוא בגודל 700. לכן הזרימה המכסימלית בו היא בגודל 700. אם נזרים מ s ל t_1 זרימה בגודל 700 אז כל הזרימה עוברת דרך t.

יהא $G(V,E)$ גרף מכוון, ויהיו $s,t \in V$. תארו אלגוריתם יעיל ככל האפשר, שבודק אם קיים מסלול __לא פשוט__ מ- s ל- t. הוכיחו את נכונות האלגוריתם ונתחו את סיבוכיותו.

__סיבוכיות__: $O(|V|+|E|)$

__אלגוריתם__:

1. באמצעות שתי הרצות של אלגוריתם DFS נמצא את רכיבי הקשירות החזקה בגרף.
2. באמצעות אלגוריתם BFS (או DFS) נמצא את קבוצת הצמתים אליהם יש מסלול מהצומת s.
3. נסתכל על הגרף שקבוצת קשתותיו היא הקשתות בעלות כיוון הפוך מקשתות הגרף המקורי. בגרף

זה נמצא את קבוצת הצמתים אליהם יש מסלול מהצומת t (שוב באמצעות אלגוריתם BFS).

4. התשובה היא חיובית אם ורק אם יש צומת שכלול גם בקבוצה שנמצאה בשלב 2 וגם בקבוצה

שנמצאה בצעד 3 ושהוא נמצא ברכיב קשירות חזקה שבו יותר מצומת אחד.

__הוכחה__: כדי שיהיה מסלול לא פשוט מ- s ל- t, צריך שיהיה מעגל שאליו יש מסלול מ- s וממנו יש מסלול ל- t.

שני צמתים נמצאים על מעגל משותף אם"ם הם באותו רכיב קשירות חזקה. קיום מסלול מהמעגל לצומת t שקול לקיום מסלול מהצומת t למעגל בגרף שבו הפכנו את כיוון הקשתות.

נתונה רשת זרימה מכוונת $G(V,E)$ עם קיבולים $c:E \to R^+$, ונתונה זרימה מקסימלית f. תאר/י אלגוריתם יעיל ככל האפשר שבודק האם קיימת קשת ברשת שהגדלת קיבולה מאפשרת את הגדלת הזרימה.

__סיבוכיות__: $O(|V|+|E|)$

__אלגוריתם__:

נבדוק אם קיים מסלול מצומת המקור לצומת היעד שמשתמש בקשת אחת שהיא רוויה ובמספר כלשהו של קשתות שאינן רוויות. נבצע סריקה דומה לסריקה של BFS, אך צומת יוכל להכנס לתור העתידים לסריקה לא רק פעם אחת אלא לפעמים פעמיים. לגבי כל צומת שאליו נגיע, נסמן אם הגענו אליו בלי שימוש בקשתות רוויות או במסלול שעובר גם בקשת רוויה אחת. לא נכניס לתור צמתים שאליהם הגענו על-ידי שימוש ביותר מקשת רוויה אחת. אם צומת מסומן ככזה שהגענו אליו רק תוך שימוש גם בקשת רוויה וכעת הגענו אליו מבלי שימוש בכזאת, אז נכניס אותו פעם נוספת לתור.

__הוכחת נכונות__:

אם אין מסלול שמשתמש רק בקשת רוויה אחת, אז צריך להגדיל את הקיבול של לפחות שתי קשתות כדי לקבל מסלול הוספה. אם יש מסלול שמשתמש רק בקשת אחת כזאת אז די להגדיל את הקיבול של

קשת זאת. נניח שיש מסלול מתאים באורך סופי, לצמתים שאליהם יש מסלול מתאים באורך 1 בודאי הסריקה תגיע. נניח שלכל צומת שאליו יש מסלול מתאים באורך k מגיעים וגם מגיעים בלי שימוש בקשתות רוויות אם קיים מסלול כזה, אז הצמתים שבמרחק k יכנסו לתור, תוך שמסמנים אם השתמשנו בקשת רוויה. לכן גם לצמתים שאליהם יש מסלול באורך $k+1$ נגיע תוך שנסמנם נכון.

יהא $G=(V,E)$ גרף לא מכוון עם פונקצית משקל אי שלילית $w:E\rightarrow R^+$, מוגדרת על קשתותיו.

תאר/י אלגוריתם יעל ככל האפשר למציאת קבוצת קשתות E' שסכום משקליהן מינימלי כך שבגרף שמתקבל מ- G ע"י זריקת קשתות E' אין מעגלים.

<u>סיבוכיות</u>: $O(|V|+|E|\log(|V|))$ ראו גם הערה בסוף הפתרון של השאלה.

<u>אלגוריתם</u>:

בכל רכיב קשירות נריץ על הגרף אלגוריתם למציאת עץ פורש בעל משקל מכסימלי. אוסף הקשתות E' יהיה אוסף הקשתות שלא יכללו בעץ זה. אלגוריתם למציאת עץ פורש מכסימלי יהיה למשל האלגוריתם של קרוסקל למציאת עץ פורש מינימלי שרץ על הגרף שבו לקשת בעלת משקל a בגרף המקורי יינתן משקל $M-a$ כאשר M הוא המשקל המכסימלי של קשת בגרף המקורי.

<u>הוכחת נכונות</u>: צריך למעשה להביא למכסימום את סכום משקל הקשתות שאינן מוסרות. מכיון שהמשקולות הן אי שליליות אז כל תוספת של קשת לקבוצת הקשתות הלא מוסרות, מגדילה את משקלה. לכן כל עוד אין בה מעגלים כדאי להוסיף לה קשתות. לכן בכל רכיב קשירות נקבל עץ פורש בעל משקל מכסימלי. עץ פורש בעל משקל W ברכיב קשירות בעל n צמתים בגרף המקורי הוא בעל משקל $(n-1)M-W$ בגרף שבו ביצענו את הטרנספורמציה המוזכרת על משקלי הקשתות. לכן הבעיה שקולה למציאת עץ פורש בעל משקל מינימלי לאחר ביצוע הטרנספורמציה.

<u>הערה</u>: ניתן לפתור בסבוכיות $O(|V|+|E|\log(|E|))$ אם מתייחסים רק לגרף שקבוצת צמתיו הם צמתים שנוגעים בקשתות. ניתן גם למעשה לקבל סיבוכיות שהיא המינימום בין זו לבין זו של אלגוריתם פרים.

יהא A מערך של n מספרים ממשיים. תאר/י אלגוריתם יעיל ככל האפשר למציאת תת-קבוצה של איברי A שסכומה מקסימלי, תחת האילוץ שאסור לבחור לתת-קבוצה זאת שני איברים סמוכים מ-A.

<u>סיבוכיות</u>: $O(n)$

<u>אלגוריתם</u>: נשתמש בתכנות דינמי.

יהיו $a[k]$ אברי המערך. נגדיר $F(n+1)=F(n+2)=0$, עבור כל $1\le k\le n$ נחשב את $F(k)$ באופן הרקורסיבי הבא:

$F(k)=\max\{F(k+1),a[k]+F(k+2)\}$

. לאחר שנחשב את כל הערכים של F , נעבור על פני איברים החל מהראשון כשנגיע לאבר k נבחר אותו אם"ם מתקיים $a[k]+F(k+2)\ge F(k+1)$ ובמקרה זה נעבור ל $k+2$, אם לא נבחר אותו אז נעבור ל $k+1$.

<u>הוכחת נכונות</u>: בהינתן ערכי הפתרונות האופטימלים עבור כל הקטעים בין p ל n עבור כל $p>k$, אז הפתרון האופטימלי בקטע בין k ל p אומר שיש לבחור את $a[k]$ ולדלג על האיבר שבמקום ה-$k+1$ או לקבל את הפתרון האופטימלי החל ממקום $k+1$.

יש עבור כל k מספר סופי קבוע של השוואות.

נתון גרף לא מכוון $G=(V,E)$, עם צבע $c(e)$ (לבן או שחור) לכל קשת e , ומשקל שלם $1\le w(e)\le 100$ לכל קשת e , כשהיצוג ע"י רשימות שכנות ומשקל כל קשת וצבעה מופיעים ליד הופעותיה ברשימות השכנות. תאר/י אלגוריתם יעיל ככל האפשר המוצא מבין כל העצים הפורשים של G המכילים את המספר הגדול ביותר האפשרי של קשתות לבנות, עץ כזה בעל משקל מינימלי.

<u>סיבוכיות</u>: $O(|V|+|E|)$

<u>הסבר</u>:

נפעיל את האלגוריתם של פרים למציאת עץ פורש מינימלי על גרף שבו משקלי הקשתות יהיו פונקציה של משקלי הקשתות בגרף המקורי ושל צבען. קשת לבנה תקבל את משקלה המקורי וקשת שחורה תקבל את משקלה המקורי ועוד $200|V|$ (בחירה שרירותית). כך תהיה עדיפות מוחלטת למספר הקשתות הלבנות ומבין העצים בעלי משקל שווה של קשתות לבנות יועדפו בעלי המשקל הנמוך. את הצמתים שאליהם עדיין לא הגענו נחזיק ב 200 תורים שכל אחד מהם יכלול קבוצה של צמתים בעלי מרחק מינימלי מסוים לקבוצת הצמתים שכבר צברנו במהלך ביצוע האלגוריתם, צומת יוכל לעבור רק לתורים המציינים מרחק נמוך יותר. בכל שלב נצרף לעץ צומת קרוב כמה שיותר. מספר העידכונים הכולל במשך ביצוע האלגוריתם לגבי צמתים שעדיין לא בעץ הוא $O(|e|)$. כל עידכון לוקח לא יותר ממספר קבוע של פעולות.

<u>הערה</u>: סיבוכיות האלגוריתם של קרוסקל לא תהיה כאן לינארית. אמנם בעזרת radix sort אפשר למיין את הקשתות בזמן לינארי, אך הסיבוכיות של union-find אינה לינארית.

נתון גרף מכוון $G(V,E)$, ולכל קשת e זוג משקלים $(a(e),b(e))$, שערכו $(0,0)$, $(0,-1)$ או $(-1,0)$.

תארו אלגוריתם יעיל ככל האפשר שבודק האם קיים ב- G מעגל מכוון (לא בהכרח פשוט) בך שסכום רכיבי ה- a שלו שלילי וגם סכום רכיבי ה- b שלו שלילי. הוכיחו את נכונות האלגוריתם וחשבו את סיבוכיותו.

סיבוכיות: $O(|V|+|E|)$

תאור האלגוריתם

על-ידי שתי הרצות של אלגוריתם DFS נמצא את אוסף רכיבי הקשירות החזקה. בשלב הבא נעבור על פני כל קשתות הגרף ולגבי כל קשת נייחס אותה לרכיב קשירות חזקה מסוים אם שני קצוותיה באותו רכיב.

אחר-כך נעבור על-פני רכיבי הקשירות החזקה השונים ונבדוק אם קיים רכיב כזה שבו גם קשת מסוג $(0,-1)$ וגם קשת מסוג $(-1,0)$. אם"ם התשובה לכך היא חיובית אז קיים מעגל כנדרש.

הוכחת נכונות

שתי קשתות שאינן ברכיב קשירות חזקה אחד אינן על מעגל כי יכול להיות מסלול לכל היותר מצמתי אחת משתי הקשתות לצמתי הקשת האחרת. ברכיב קשירות חזקה יש מסלול מכוון מכל צומת לכל צומת, לכן יש מעגל שעובר בכל הצמתים שבקצוות של שתי קשתות.

יהא $G(V,E)$ גרף לא מכוון וקשיר, עם משקלים $w:E\to R^+$.

א. תארו אלגוריתם יעיל ככל האפשר אשר בהינתן צומת $u\in V$ מוצא מבין העצים הפורשים

המינימלים עץ T שעבורו $d_T(u)$ מקסימלי, כאשר $d_T(u)$ זו דרגת u בעץ T. הוכיחו את נכונות

האלגוריתם ונתחו את סיבוכיותו.

סיבוכיות: $O(|E|\log(|V|))$ ראו גם הערה בסוף הפתרון של השאלה.

אלגוריתם:

1. נמיין את קשתות הגרף לפי משקליהן.
2. נעבור על פני הקשתות כפי שמוינו.נמצא את d-הערך המוחלט של ההפרש המינימלי בין שתי קשתות בעלות משקלות שונים בגרף.
3. נקבע משקלות חדשים לקשתות הגרף: את המשקלות של קשתות שלא נוגעות בצומת u לא נשנה.

לגבי כל קשת שנוגעת בצומת u, נחסיר ממשקלה $\frac{d}{2}$.

4. נריץ את האלגוריתם של קרוסקל על הגרף עם המשקלים החדשים.

הוכחה:

נסתמך על הנכונות של האלגוריתם של קרוסקל. כאשר בכל שלב מצרפים לקבוצת צמתי העץ קשת שאינה כבדה יותר מאף לא קשת שעדיין לא עברנו עליה ושלא יוצרת מעגל עם הקשתות שכבר צרפנו, אז מתקבל עץ פורש מינימלי. בהינתן שעד שלב מסוים בחרנו רק קשתות שהן חלק מעץ אופטימלי גם בגרף המקורי, אז קשת שתצטרף באותו שלב, יכלה גם להצטרף לפתרון בגרף המקורי. (קשת שנבחרת והיא לא כבדה יותר מאחרת שעדיין לא הגענו אליה היתה גם לא כבדה יותר בגרף המקורי והאלגוריתם של קרוסקל לא מסתכל בכמה היא היתה קלה יותר.)

מבין העצים שהיו בגרף המקורי בעלי משקל זהה, יהיה כעת משקל נמוך יותר לבעל יותר שכנים ל- u.

הסיבוכיות היא של האלגוריתם של קרוסקל שחלק ממנו עצמו הוא ביצוע מיון לקשתות.

ב. תארו אלגוריתם יעיל ככל האפשר אשר בהינתן שני צמתים $u,v\in V$ מוצא מבין העצים הפורשים

המינימלים עץ T שעבורו $d_T(u)-d_T(v)$ מקסימלי. הוכיחו את נכונות האלגוריתם ונתחו את סיבוכיותו.

סיבוכיות: $O(|E|\log(|V|))$ ראו גם הערה בסוף הפתרון של השאלה.

אלגוריתם:השינוי היחיד שנכניס לאלגוריתם של סעיף א' הוא בשלב 3. כאן לכל קשת שנוגעת בצומת u נחסיר ממשקלה $\frac{d}{3}$ ולכל קשת שנוגעת בצומת v נוסיף למשקלה $\frac{d}{3}$. אם יש קשת בין u ל v אז לא נשנה את משקלה.

הוכחה:

שוב קשת שנבחרת והיא לא כבדה יותר מאחרת שעדיין לא עברנו עליה גם לא כבדה יותר מהאחרת בגרף המקורי. מבין העצים האופטימלים בגרף המקורי יהיה משקל מינימלי לבעל הפרש דרגות מכסימלי בין u ל- v.

הערה לגבי שני הסעיפים: ניתן גם לפתור בסיבוכיות של האלגוריתם של פרים.

תחילה נסיר מהגרף את הצומת או שני הצמתים המיוחדים. באמצעות האלגוריתם של פרים נחשב עץ פורש מינימלי בגרף שנותר. אחר-כך נוסיף את הצומת או את שני הצמתים המיוחדים ונפתור כמו בשאלה הראשונה בקובץ זה על-פי האלגוריתם של קרוסקל כאשר נותנים עדיפות חיובית או שלילית כפי שתארתי לקשתות שנוגעות בצמתים המיוחדים ומספר הקשתות הוא בסדר גודל של מספר הצמתים.

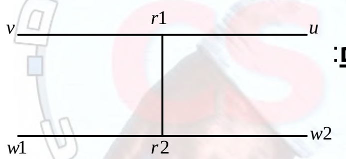
זו דרך עקיפה. אבל ניתן לפתור גם מראש בעזרת גירסא של האלגוריתם של פרים.

יהא $G(V,E)$ עץ לא מכוון. המרחק בין שני צמתים ב- G הוא אורך המסלול הקצר ביותר ביניהם, בספירת קשתות. המרחק המכסימלי בין שני צמתים כלשהם בגרף הוא הקוטר של הגרף. תנו אלגוריתם מהיר ככל שתוכלו שמוצא את הקוטר של G. מה סיבוכיות האלגוריתם שנתתם ?

(רמז: יהא v צומת כלשהו בגרף. מה תוכלו לאמור על צומת u שמרחקו מ- v הוא מכסימלי?)

סיבוכיות: $O(|V|)$

אלגוריתם:



1. נבחר צומת v שרירותי בעץ. מצומת זה נחשב באמצעות אלגוריתם BFS את המרחקים לכל צמתי העץ. נבחר צומת u שעבורו התקבל מרחק כזה מכסימלי.

2. מהצומת u נחשב באמצעות אלגוריתם BFS את המרחקים לכל צמתי העץ. המרחק המכסימלי שיתקבל בשלב הזה הוא קוטר העץ.

נכונות: צריך למעשה להוכיח שהצומת u הוא קצה של קוטר.

נניח בשלילה ש u אינו קצה של קוטר. אז קיימים זוג צמתים $w1$ ו $w2$ שהמרחק ביניהם גדול מהמרחק של u מכל צומת בעץ.

שני המסלולים מ-v ל-u ומ-$w1$ ל-$w2$ מחוברים על-ידי מסלול מ-$r1$ ל-$r2$. (יתכן ש $r1$ ו $r2$ הם אותו צומת.) המרחק מ-u ל-$r2$ קטן מהמרחק מ-$w1$ ל-$r2$. לכן המרחק מ-u ל-$r1$ קטן מהמרחק מ-$w1$ ל-$r1$. לכן המרחק מ-v ל-u קטן מהמרחק מ-$w1$ ל-v. לכן u לא היה יכול להבחר כבעל מרחק מכסימלי מ-v בשלב 1.

מספר הקשתות בעץ הוא $|V|-1$, לכן הסיבוכיות של BFS היא כאן $O(|V|)$.

נתון גרף לא מכוון $G=(V,E)$, מיוצג ע"י רשימות שכנות. תאר/י אלגוריתם יעיל ככל האפשר שיקבע אם קיים תת גרף $H=(V,E')$ של G, שקבוצת צמתיו היא קבוצת הצמתים של G, ו $E' \subseteq E$ מקיימת $|E'| \le 2|V|$, כאשר H עצמו גרף קשיר וחסר גשרים (ז"א: לכל $e \in E'$ גם הגרף H ללא הקשת e הוא קשיר).

סיבוכיות: $O(|V|+|E|)$

הסבר:

באמצעות אלגוריתם DFS נבדוק אם הגרף G הוא גרף קשיר וחסר גשרים. אם הוא לא קשיר או בעל גשרים, אז בודאי אם נסיר ממנו קשתות הוא יהיה בלתי קשיר או בעל גשרים. אם הוא קשיר וחסר גשרים אז יש לו עץ פורש בעל $|V-1|$ קשתות. נראה שניתן להוסיף לעץ לא יותר מ $|V-1|$ קשתות ולקבל גרף קשיר וחסר גשרים. לצומת שהוא עלה בעץ יש קשת שהיא לא קשת בעץ שמחברת אותו לצומת אחר בגרף. נצרף את הקשת הזאת ובכך יווצר תת גרף חסר גשרים שכולל את העלה. כעת בכל שלב שבו עדיין יש צומת שאינו בתת הגרף חסר הגשרים של העלה, נוסיף קשת שמחברת את רכיב זה עם צומת שאינו ברכיב זה ובכך ירד מספר הצמתים שאינם בתת גרף חסר הגשרים של העלה בלפחות 1. בתחילה היה מספר צמתים אלה שווה ל $|V-1|$ לכן דרושות לא יותר מ $|V-1|$ תוספות של קשתות כדי להורידו לאפס.

נתונה רשת זרימה בה הקיבולים של כל הקשתות הם מספרים שלמים וזוגיים, מלבד קשת אחת (u,v) שלה קיבול אי-זוגי. נתונה זרימה מקסימלית f , ונניח שערכה אי-זוגי. האם (u,v) בהכרח רוויה ?

תשובה: כן

הוכחה:

נניח בשלילה שהיא לא רוויה. אם היא לא רוויה אז ניתן להקטין את קיבולה ועדיין לקבל את אותה זרימה מקסימלית. ערך זרימה זאת הוא אי-זוגי. אך זרימה מקסימלית שווה לגודל חתך מינימלי ,

לאחר השינוי ישארו קיבולי כל יתר הקשתות זוגיים וקיבול קשת זאת לא יהיה מספר שלם אי-זוגי לכן גודל החתך המינימלי לא יהיה מספר אי-זוגי.

נתונה סדרת מספרים ממשיים $X_1, X_2, \cdots, X_n$. תארו אלגוריתם יעיל ככל האפשר המחשב לכל אינדקס j את מספר התת-סדרות העולות ממש המסתיימות ב- x_j (תת-סדרה עולה ממש היא תת-סדרה $X_{a_1}, X_{a_2}, \cdots, X_{a_i}$, כאשר $a_1 < a_2 < \cdots < a_i$, המקיימת $X_{a_1} < X_{a_2} < \cdots < X_{a_i}$. היא מסתיימת ב- x_j אם $a_i = j$.)

יעילות: $O(n^2)$

אלגוריתם: נשתמש בתכנות דינמי. עבור המספר הראשון בסדרה מספר התת-סדרות המתאימות הוא 1. נעבור על כל האינדקסים מונוטונית מהקטן לגדול. עבור כל איבר, מספר התת-סדרות המתאימות שמסתיימות בו יהיה שווה ל 1 ועוד הסכום של מספרי התת-סדרות המתאימות שמסתיימות במספרים בעלי אינדקס נמוך יותר שהם גם קטנים מהמספר עצמו (עוברים על כל המספרים שהם גם בעלי אינדקס נמוך יותר וגם קטנים יותר).

הסבר: עבור כל j כל תת סדרה מתאימה או שמכילה רק את המספר בעל אינדקס j או שהאיבר הלפני האחרון הוא אחד המספרים הקטנים יותר שהם בעלי אינדקס קטן יותר. זאת חלוקה למקרים זרים לפי זהות האיבר הלפני אחרון.

לגבי כל אינדקס j עוברים על פני לא יותר מ n אינדקסים קטנים יותר ומסכמים לא יותר מ n מספרים.

נתון גרף לא מכוון $G(V,E)$ מיוצג ע"י רשימות שכנות, נתון משקל שלם $1 \le w(e) \le |V|$ לכל קשת $e \in E$, ונתונים זוג צמתים $s,t \in V$. תאר/י אלגוריתם יעיל

ככל האפשר המוצא מסילה מ- s ל- t כך שהמשקל המירבי של קשת במסילה הוא קטן ככל האפשר.

סיבוכיות: $O(|V|+|E|)$

הסבר: האלגוריתם יורכב משני שלבים עיקריים בשלב הראשון נמצא אוסף קשתות בעלות משקל מכסימלי קטן ככל היותר שמאפשרות קשירות בין s ל t. בשלב השני נבצע BFS בגרף שאוסף קשתותיו נמצא בשלב הראשון. ה- BFS יתחיל בצומת s עד הגעה לצומת t כאשר לא נסתכל על המשקל של כל קשת באוסף זאת כי כבר ברור שבשביל לגרום ל- s ול- t להיות ברכיב קשירות אחד, צריך קשתות במשקל של לפחות המשקל המכסימלי של קשתות האוסף.

השלב הראשון: נעבור על-פני קשתות הגרף ונכניס את כל אחת מהן לרשימה של קשתות בעלות משקל שווה לשלה (יהיו $|V|$ רשימות של קשתות). נסמן את הצומת s שהגענו אליו. נסמן את יתר הצמתים ככאלה שבשלב זה לא הגענו אליהם. לכל צומת נחזיק תור של קשתות שנוגעות בו ושעליהן צריך לעבור כאשר נגיע לצומת זה. בתחילה יהיו בתורים רק הקשתות בעלות משקל 1. נחזיק תור של צמתים שאליהם הגענו ושצריכים להיסרק. כל עוד לא הגענו לצומת t נסרוק צמתים שבתור זה. צומת שהתור שלו יתרוקן יצא לפחות באופן זמני מהתור של הצמתים שצריכים להיסרק. כאשר התור של הצמתים יתרוקן מבלי שהגענו לצומת t אז נוסיף קשתות חדשות לתורי הקשתות של צמתים שבקצוותיהם ומבין הצמתים האלה נוסיף את אלה שאליהם כבר הגענו לתור הצמתים שצריכים להיסרק. הקשתות החדשות שנוסיף הן הקשתות מהרשימה של הקשתות הכי קלות מבין אלה שעדיין לא עברנו עליה. צמתים שאליהם נגיע לראשונה במהלך הסריקה גם יכנסו לתור הצמתים שאותם צריך לסרוק (לגבי כל צומת כזה, יתכן שכבר יש בתור הקשתות שלו קשתות שעליהן צריך לעבור, הקשתות האלה נצברו כבר לפני שהגענו לצומת זה).

הערה: ניתן לפתור גם בדרכים שונות לחלוטין.

נתון גרף מכוון $G=(V,E)$, המיוצג ע"י רשימות שכנות.

תארו אלגוריתם יעיל ככל האפשר, המחשב את הקבוצה $U\subseteq V$ של הצמתים u, בעלי התכונה שלכל $v\in V$ קיימת מסילה מכוונת מ- u ל- v.

יעילות: $O(|E|+|V|)$

אלגוריתם: באמצעות שתי הרצות של אלגוריתם DFS נמצא את רכיבי הקשירות החזקה בגרף.

נחשב את גרף העל של הגרף G.

לגבי גרף העל, נסתכל על הגרף שאוסף קשתותיו הן בעלות כיוון הפוך לקשתות הגרף המקורי, בגרף זה נבחר צומת שרירותי וממנו נעשה BFS לצורך מציאת עלה- w.

בגרף העל המקורי נבצע BFS מהצומת w. אם"ם בסריקה זאת נגיע לכל צמתי הגרף, אז צמתי רכיב הקשירות שמייצג w הם הקבוצה U. אחרת הקבוצה ריקה.

הסבר: מכל צומת בקבוצה U צריך להיות מסלול לכל צומת אחר בקבוצה U. לכן קבוצת הצמתים U היא רכיב קשירות חזקה. בגרף שכיוון קשתותיו הפוך לכיוון המקורי דרוש שיהיה מסלול מכל צומת לצמתי U ושמצמתי U לא יהיה מסלול לצמתים אחרים כי אחרת הם היו חלק מאותו רכיב קשירות חזקה. לכן

BFS בגרף שכיוון קשתותיו הפוך לכיוון המקורי של קשתות גרף העל חייב להגיע לצומת שמייצג את הקבוצה U ושזה יתגלה כעלה.

נתונה מחרוזת T של n תווים ותבנית P של m תווים. תארו אלגוריתם יעיל ככל האפשר הבודק אם קיימות מחרוזות x,y ורישא (לא ריקה) P' של P, כך ש- $T=xP'y$ וגם מתקיים $|y|<2|P'|$

יעילות: $O(\min(m,n))$

אלגוריתם: נתייחס לתבנית שהיא התבנית המקורית אם $n\ge m$ ואחרת היא רק n התווים הראשונים בתבנית המקורית. נריץ וריאציה של אלגוריתם KMP למציאת כל ההופעות של התבנית במחרוזת חלקית של T שהיא $\min(n,3m)$ התווים האחרונים של T : במקום שבו תתגלה התאמה מלאה נזכור אותה ובנוסף נתייחס גם למקרה שלכאורה לא היתה התאמה במקום האחרון של התבנית לאחר שעד מקום זה היתה התאמה. כך נקבל את ההתאמה הארוכה ביותר שמסתיימת באותו מקום מבין ההתאמות הקצרות מההתאמה המוחלטת וימשך תהליך מציאת התאמה ימנית יותר. עבור כל מקום k במחרוזת T המקורית, נקבל את $\max_k$ - אורך הרישא הארוכה ביותר שמסתיימת בו. הדרישה מתקיימת אם"ם קיים k כך ש- $2\max_k > n-k$.

הסבר: אם $n<m$ אז אם יש התאמה של n התווים הראשונים אז זאת כבר ההתאמה הארוכה ביותר האפשרית. אם יש התאמה של רישא מסוימת במקום מסוים, אז תמיד אפשר לבחור את x ואת y שיהיו זהים לתתי המחרוזות שלפני הרישא המתאימה ואחרי הרישא המתאימה. קיום k כך ש $2\max_k > n-k$ הוא בדיוק הדרישה הנחוצה. באלגוריתם KMP הערכים $\max_k$ תמיד מחושבים עד מציאת התאמה. הדרישה שיתקיים $|y|<2|P'|$ אומרת שההתאמה החלקית או המלאה חייבת להיות ב $3m$ המקומות האחרונים של T .

לגבי היעילות: כל הטיעונים לגבי מספר ההשוואות באלגוריתם המקורי תקפים גם כאן. בפרט פרוצדורת ההכנה של התבנית לא משתנה. אורך המחרוזת שעליה אנו עושים את הבדיקה הוא $\min(n,3m)$.

נתון גרף לא מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות, וצומת $s\in V$. לכל קשת $e\in E$ יש משקל שלם $w(e)$ (חיובי או שלילי). תארו אלגוריתם יעיל ככל האפשר המחשב, לכל צומת $v\in V$, את המשקל המינימלי של מסלול מ- s ל- v בעל מספר זוגי של קשתות. (הניחו שב- G אין מעגלים שליליים.)

יעילות: $O(|E|+|V|\log|V|)$ כמו דיקסטרה או אם לא מניחים שהקשתות אי-שליליות אז $O(|V||E|)$ כמו בלמן פורד

הערה: הגרף הוא גרף לא מכוון ואין מעגלים בעלי אורך שלילי, לכן ניתן להסיק שהקשתות כולן הן בעלות משקל אי-שלילי.

אלגוריתם: נסתכל על גרף דו-צדדי שבו לכל צומת מהגרף המקורי יש עותק בכל צד. עבור כל זוג צמתים i,j משני צדדים שונים, יש בינהם קשת אם בגרף המקורי יש קשת בין שני הצמתים שהם מייצגים. משקל קשת זאת זהה למשקל הקשת מהגרף המקורי. בגרף זה נמצא את המרחקים מעותק אחד של הצומת s לכל יתר הצמתים שבצד זה. אלה המרחקים הדרושים.

הסבר: מכיון שהגרף שנבנה הוא דו-צדדי אז כל המסלולים בעלי אורך זוגי שמתחילים בצד אחד מסתיימים באותו צד. לכל מסלול בעל אורך זוגי בגרף המקורי, יש מסלול מתאים בגרף זה. כל מסלול בגרף זה מייצג מסלול בגרף המקורי.

מכיון שבגרף המקורי אין מעגלים בעלי אורך שלילי, אז גם בגרף הדו-צדדי אין מעגלים בעלי אורך שלילי. מספר צמתי הגרף הוא כפליים מספר צמתי הגרף המקורי ומספר קשתות הגרף הוא כפליים מספר קשתות הגרף המקורי. מכאן מתקבלת היעילות הרשומה.

נתון גרף מכוון $G(V,E)$, המיוצג ע"י רשימות שכנות.

תארו אלגוריתם יעיל ככל האפשר, המוצא קבוצת צמתים לא ריקה $U\subseteq V$ מגודל מינימלי אפשרי, כך שאין שום קשת מכוונת היוצאת מצומת של U לצומת שאינו ב- U.

יעילות: $O(|V|+|E|)$

אלגוריתם:

1. באמצעות שתי הרצות של אלגוריתם DFS נמצא את רכיבי הקשירות החזקה בגרף.

2. נמצא את כל רכיבי הקשירות החזקה שמהם לא יוצאות קשתות לצמתים שמחוץ להם.

3. לקבוצה U נבחר את כל צמתי הרכיב בעל מספר מינימלי של צמתים מבין הרכיבים שנמצאו בשלב 2.

הסבר: מכל צומת יש מסלול לכל צומת שבאותו רכיב קשירות חזקה. לכן אם בוחרים צומת מסוים, אז כל הצמתים מרכיב הקשירות החזקה שלו צריכים להבחר. מכל רכיב קשירות חזקה יש מסלול לרכיב קשירות חזקה שמימנו אין קשתות אל מחוצה לו. לכן חייבים לבחור לפחות רכיב קשירות חזקה אחד שממנו אין יציאות. בחרנו רכיב כזה שהוא בעל גודל מינימלי.

==

יהא $G=(V,E)$ גרף לא מכוון מיוצג ע"י רשימות שכנות, עם משקל שלם $1\le w(e)\le|E|$ לכל קשת $e\in E$ כך שאין אף זוג קשתות בעלות אותו משקל.

קשת נקראת <u>כבדה</u> אם היא בעלת משקל מקסימלי במעגל פשוט כלשהו ב- G.

תארו אלגוריתם יעיל ככל האפשר שימצא את כל הקשתות הכבדות בגרף.

יעילות: $O(|V|+|E|\cdot\alpha(|E|,|V|))$

אלגוריתם: באמצעות הרצה בודדת של האלגוריתם של קרוסקל על כל הגרף נמצא עצים פורשים מינימלים בכל רכיבי הקשירות של הגרף. הקשתות הכבדות הן הקשתות שלא יבחרו ליער זה.

הסבר: כאשר כל המשקולות שונים, הקשתות שלא נבחרות כאשר מריצים את אלגוריתם קרוסקל הן הקשתות שסוגרות מעגל עם קשתות בעלות משקל קטן ממשקלן.

במקרה המתואר ניתן למיין את הקשתות לפי אורכן בזמן $O(|E|+|V|)$. לכן כאן השלב שצורך יותר זמן הוא השלב השני באלגוריתם של קרוסקל.

נתון גרף מכוון $G(V,E)$, המיוצג ע"י רשימות שכנות עם משקל שלם $w(e)$ (חיובי או שלילי) לכל קשת $e\in E$. נתון שאין ב- G מעגל שלילי.

תארו אלגוריתם יעיל ככל האפשר המוצא לכל $v\in V$ את המשקל הקטן ביותר של מסילה מכוונת (בעלת מספר קשתות כלשהו) המתחילה ב- v.

יעילות: $O(|E||V|)$

אלגוריתם: נמצא את הגרף בעל אותה קבוצה של צמתים ובעל קשתות בכיוון הפוך מהקשתות המקוריות.

לגרף זה נוסיף את צומת s וממנו נוסיף קשתות בעלות משקל אפס לכל אחד מצמתי הגרף.

בגרף שהתקבל נמצא באמצעות אלגוריתם בלמן פורד את משקלי המסילות הקצרות ביותר מהצומת s לכל אחד מצמתי הגרף.

לגבי כל צומת, המרחק שהתקבל מהצומת s בגרף שבנינו, הוא המרחק המינימלי מצומת זה בגרף המקורי.

אם רוצים להימנע ממסילות ריקות ללא קשתות אז משכפלים שני עותקים של צמתי הגרף, הקשתות בכיוון הפוך יהיו בין הנציגים של הצמתים שבאותו עותק וגם בין צמתי העותק הראשון לצמתי העותק השני (רק בכיוון זה). מהצומת s יצאו הקשתות שאותן הוספנו רק לנציגי העותק הראשון.

המרחק לצומת יוגדר כמרחק לנציג שלו בעותק השני.

<u>הסבר</u>: המרחק מצומת s לצומת v בגרף שבנינו שווה למרחק הכי קטן מצומת אחר לצומת v.

זה שווה למרחק הכי קטן מצומת v לאיזשהו צומת בגרף המקורי. על-ידי היפוך כיוון הקשתות והוספות רק קשתות יוצאות מהצומת s לא יצרנו מעגלים בעלי אורך שלילי (לא ניצור כלל מעגלים חדשים). גם אם נשכפל את צמתי הגרף לשני עותקים לא ניצור מעגלים חדשים. המסילות לצמתי העותק השני עוברות לפחות קשת אחת בגרף המקורי.

נתון גרף מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות ונתונה קבוצת צמתים $U\subseteq V$. תארו אלגוריתם יעיל ככל האפשר שיקבע אם יש מסילה מכוונת (לאו דווקא פשוטה; קרי ייתכן שתעבור בצמתים וקשתות יותר מפעם אחת) שמבקרת בכל הצמתים ב- U.

<u>יעילות</u>: $O(|V|+|E|)$

<u>אלגוריתם והסבר</u>:

נחשב את רכיבי הקשירות החזקה. נמצא מיון טופולוגי של גרף העל.

בגרף העל נבצע DFS החל מהרכיב שממנו יש מסלולים לכל יתר הרכיבים.

יש מסלול כנדרש אם"ם כל רכיבי הקשירות החזקה שבהם יש ייצוג לצמתים מהקבוצה U הם על מסלול פשוט בסריקה שמצאנו. אם אין מסלול כזה אז זה אומר שבין לפחות שני צמתים מ U אין מסלול באף לא כיוון כי ברור שבכיוון ההפוך לסריקה אין מסלול כזה. בכל רכיב קשירות חזקה יש מסלול מכל צומת לכל צומת ובחזרה ולכן ניתן גם לצאת מכל צומת שלו אל רכיב קשירות חזקה אחר. בסריקה אפשר לקבל סדרים שונים. כדי להתגבר על זה נבדוק אם מכל צומת שמייצג צומת שבו צמתים מיוחדים יש מסלול לצומת שהוא הבא אחריו במיון הטופולוגי שבו צמתים מיחדים. אם עבור כל שני זוגות עוקבים זה מתקיים אז יש מסלול כנדרש. נדאג שנגיע לצומת הבא לפני שנסוגונו סופית מהקודם.

נתון גרף לא מכוון דו-צדדי $G=(V,E)$ המיוצג ע"י רשימות שכנות.

תארו אלגוריתם יעיל ככל האפשר המוצא ב- G תת גרף בעל מספר מירבי של קשתות, שבו לכל קודקוד דרגה לכל היותר 3.

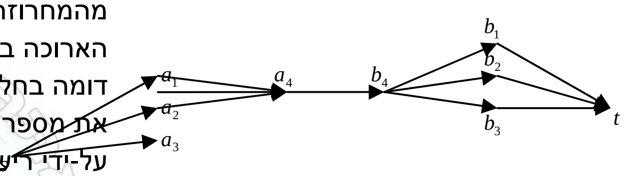
<u>יעילות</u>: $O\left(|E||V|^{2/3}\right)$

<u>אלגוריתם</u>:

נבנה רשת זרימה שעליה נפעיל אלגוריתם זרימה.

לכל צומת מהגרף המקורי יהיו ארבעה צמתים נציגים. נוסיף שני צמתים s ו t לגרף. יהיו קשתות מכוונות מ s לשלושה עותקים של כל צומת בצד הראשון. תהיה קשת מכל אחד מנציגים אלה אל הנציג הרביעי של אותו צומת. יהיו קשתות מכוונות משלושה נציגים של כל צומת מהצד השני אל הצומת t. מכל נציג רביעי יהיו קשתות מכוונות לשלושת הנציגים האחרים של אותו צומת. עבור כל זוג צמתים שיש ביניהם קשת בגרף המקורי תהיה קשת מכוונת מהנציג הרביעי של הצומת שבצד הראשון אל הנציג הרביעי של הצומת שבצד השני. לכל הקשתות יהיה קיבול של יחידה אחת. נפעיל את אלגוריתם דיניץ אלגוריתם למציאת זרימה מכסימלית מ s ל t. זוגות הצמתים מהגרף המקורי שביניהם תהיה קשת הם הצמתים שבין הנציגים הרביעיים שלהם יש זרימה בגרף ברשת שבנינו.

<u>הערה</u>: נייצג ברשת רק צמתים שלהם דרגה לפחות 1 בגרף המקורי.



בשרטוט נתונה רשת שמייצגת בניה לפי גרף מקורי שבו שני צמתים a ו b שביניהם יש קשת.

<u>הסבר</u>: בכל צומת יכולה לעבור זרימה של לכל היותר שלוש יחידות. הקישור דרך הנציגים הרביעיים דואג לכך שלא תהיה זרימה של יותר מיחידה אחת בין שני צמתים. זאת היא רשת זרימה שבה מספר הצמתים הוא באותו סדר גודל של מספר הצמתים שבגרף המקורי ומספר הקשתות הוא בסדר גודל של מספר הקשתות שבגרף המקורי (לא יצגנו צמתים שלהם דרגה אפס בגרף המקורי). זאת היא רשת זרימה עם קיבולים של יחידה אחת וללא קשתות מקבילות.

נתונה מחרוזת T של n תווים ותבנית P של m תווים, $|T|=n$, $|P|=m<n$.

תארו אלגוריתם יעיל ככל האפשר הבודק האם קיים k שלם כך ש-

$$T=P_{i_1}P_{i_2}\ldots P_{i_k}$$

כאשר כל P_{i_j} הוא רישא של P, ואם קיים k כזה הוא מוצא את <u>ה-k הקטן ביותר</u> עבורו זה מתקיים.

<u>יעילות</u>: $O(n)$

<u>אלגוריתם</u>: <u>השלב הראשון</u>: נריץ את אלגוריתם KMP למציאת כל ההופעות של התבנית במחרוזת. גם אם במקום מסוים קבלנו התאמה מלאה אז נמשיך בחיפוש התאמה בהמשך כאילו היה כשלון במקום האחרון בתבנית. כך נקבל עבור כל מקום במחרוזת את אורך הרישא הארוכה ביותר של P שמסתיימת בו.

<u>השלב השני</u>: אם במקום האחרון במחרוזת מסתיימת רישא, אז נבחר את הרישא הארוכה ביותר המסתיימת במקום זה. נקצץ מהמחרוזת את הסיפא שחופפת לרישא הארוכה ביותר שנמצאה. נמשיך בתהליך דומה בחלק של המחרוזת שנותר. נספור את מספר הפעמים שנקצץ סיפא. הכיסוי על-ידי רישות אפשרי אם באיזשהו שלב נגיע למצב שקיצצנו כבר את כל המחרוזת.

<u>הסבר</u>: מבלי לפגוע בקיומו של פתרון וגם כדי להביא למינימום את מספר הרישות שבכיסוי, ניתן לבחור בכל שלב ברישא הארוכה ביותר האפשרית של התבנית

שתכסה סיפא של המחרוזת. אם קיים פתרון אחר שמשתמש ברישא קצרה יותר, אז ניתן לקצץ בפתרון זה חלק מרישות קודמות, כך שנאריך את הרישא האחרונה על חשבון הקודמות ובסך הכל נכסה את אותו חלק של המחרוזת.

סיבוכיות השלב הראשון זהה לסיבוכיות של אלגוריתם KMP : המשכנו בחיפוש התאמה כאילו לא היתה התאמה במקום האחרון בתבנית. בשלב השני, כמובן לא נקצץ חלקים מהמחרוזת יותר מ n פעמים.

תארו אלגוריתם יעיל ככל האפשר שבודק, בהינתן שתי מחרוזות S,T באורך n כל אחת, האם T היא הזזה ציקלית של S (לדוגמא car ו arc הן הזזות ציקליות אחת של השניה).

יעילות $O(n)$

אלגוריתם T_2 יהיה טקסט שיורכב מרצף של שתי מחרוזות T אחת אחר השניה. נריץ אלגוריתם לבדיקה אם S מופיע ב T_2. יש הזזה ציקלית אם S יופיע ב T_2. אם מתייחסים רק להזזות ממש ולא כוללים בהן זהות בין מחרוזות, אז בודקים רק אם S מופיע בטקסט של $2n-2$ אותיות שלא כולל את האות הראשונה והאחרונה של T_2.

הסבר אם S היא הזזה ציקלית של T ב k מקומות, אז S יופיע ב T_2 החל ממקום $k+1$ (אם S ו T הן זהות אז הוא יופיע החל ממקום 1).

יהא $G=(V,E)$ גרף קשיר, לא מכוון, עם מחיר לכל קשת.

הוכח או הפרך: אם $G_1=(V,E_1)$ ו $G_2=(V,E_2)$ הם שני גרפים קשירים על אותה קבוצת צמתים V עם מחיר לכל קשת, ו $T_1=(V,F_1)$ עץ פורש מינימלי ל-G_1, $T_2=(V,F_2)$ עץ פורש מינימלי ל-G_2 אזי עץ פורש מינימלי בגרף $(V,F_1\cup F_2)$ הוא גם עץ פורש מינימלי של $G=(V,E_1\cup E_2)$. (נניח כאן כי $E_1\cap E_2=\phi$).

פתרון

נראה שהטענה נכונה. נניח שיש עץ פורש מינימלי של G שבו יש קשת e שהיא ב $E_1\cup E_2$ אך לא ב $F_1\cup F_2$. נניח בלי הגבלת הכלליות שהיא ב E_1. נראה שיש עץ פורש מינימלי אחר של G שבו היא לא נמצאת ובמקומה נמצאת קשת ששיכת ל F_1 כאשר יתר קשתות העץ לא הוחלפו. נסיר אותה מהעץ הנתון של G. בכך יווצרו שני רכיבי קשירות בגרף. הקשת e לא נכללת בעץ הפורש T_1 לכן בהכרח היא סוגרת מעגל עם קשתות מ F_1 שאף אחת מקשתות מעגל זה היא לא יותר כבדה מ e. מבין קשתות המעגל יש לפחות עוד קשת אחת שמחברת את שני רכיבי הקשירות שנוצרו עם הסרת e מהעץ של G. נוסיף את אחת מהקשתות האלה לעץ במקום e ובכך נקבל עץ שסכום משקל קשתותיו לא יותר גדול מהמקורי ולכן גם הוא עץ פורש מינימלי. כך נוכל להחליף אחת אחר השניה את כל קשתות העץ של G שאינן ב $F_1\cup F_2$ בקשתות שהן כן ב $F_1\cup F_2$.

נתון גרף מכוון $G=(V,E)$, כאשר $V=\{1,2,\cdots,n\}$, קודקוד s ב- V, ונתונה פונקצית משקל אי שלילית $w:E\to R^+\cup\{0\}$.

המסלול $\{x_0,x_1,\cdots,x_r\}$ יקרא **מונוטוני** אם $x_0<x_1<\cdots<x_r$ או $x_0>x_1>\cdots>x_r$.

תארו אלגוריתם **יעיל** למציאת משקל המסלולים **המונוטונים** הקלים ביותר מ- s לכל קודקוד.

הסבירו את נכונות האלגוריתם וחשבו את סבוכיותו.

סבוכיות $O(|E|+|V|)$

תאור האלגוריתם

שלב ראשון: נמחק מהגרף את כל הקשתות שמחברות צמתים בעלי אינדקס קטן מהאינדקס של s עם צמתים בעלי אינדקס גדול משל s או להפך.

שלב שני: נשתמש בתכנות דינאמי. המרחק לצומת s הוא אפס. עבור הצמתים בעלי אינדקס גדול מהאינדקס של s: נעבור על הצמתים האלה בסדר עולה. עבור כל צומת v כזה המרחק המבוקש יהיה המינימום על-פני המרחק לצומת שכן w בעל אינדקס קטן יותר (אך לא קטן משל s) + אורך הקשת מהצומת w ל v. עבור הצמתים בעלי אינדקס קטן מהאינדקס של s: נעבור על הצמתים האלה בסדר יורד. עבור כל צומת v כזה המרחק המבוקש יהיה המינימום על-פני המרחק לצומת שכן w בעל אינדקס גדול יותר (אך לא גדול משל s) + אורך הקשת מהצומת w ל v.

הסבר נכונות

אין מסלול מונוטוני שמתחיל בצומת s ועובר גם בצמתים בעלי אינדקס גבוה יותר וגם בצמתים בעלי אינדקס נמוך יותר. כל מסלול מ s לצומת שונה מ s עובר רק דרך צמתים בעלי אינדקס קרוב יותר לזה של s. לגבי כל צומת, בהנחה שלגבי כל הצמתים בעלי אינדקס קרוב יותר ל s חשבנו את המרחקים האופטימלים אז במעבר על כל הקשתות שנכנסות אליו נמצה את כל המסלולים הנכנסים אליו.

על כל צומת ועל כל קשת עוברים רק מספר סופי של פעמים.

נתון גרף לא מכוון, קשיר ופשוט $G=(V,E)$, פונקצית משקל $w:E\to R^+\cup\{0\}$, קודקוד s ב- V, ועץ פורש T של הגרף G.

תארו אלגוריתם **יעיל**, הבודק אם T הוא עץ מסלולים קלים ביותר מ- s לכל קודקוד בגרף. הסבירו את נכונות האלגוריתם וחשבו את סבוכיותו.

סבוכיות $O(|E|+|V|)$

תאור האלגוריתם

שלב ראשון: נחשב את המרחקים על העץ מהצומת s אל כל צמתי הגרף. נסרוק את העץ ב BFS החל מ s ולגבי כל צומת w, המרחק מ- s יוגדר כמרחק לצומת שממנו הגענו לצומת w בתוספת אורך הקשת ביניהם.

שלב שני: נעבור על פני כל קשתות הגרף. עבור כל קשת נבדוק אם קיים צומת w מבין שני קצוותיו שהמרחק על העץ לקצה השני של הקשת מ- s בתוספת אורך הקשת שביניהם קטן מהמרחק על העץ לצומת w. אם לפחות פעם אחת ימצא צומת כזה אז העץ איננו עץ מרחקים מינימלים. אחרת הוא כן.

הסבר נכונות בשלב הראשון נמצא את כל המרחקים הקצרים ביותר על העץ כי לכל צומת יש רק מסלול אחד שמוביל אליו מ- s על העץ.

לגבי השלב השני: אם קיים צומת שלגביו העץ לא נותן מרחק אופטימלי אז יש מסלול קצר יותר לצומת. במסלול זה יש צומת ראשון אליו העץ לא נותן מרחק מינימלי.כאשר נטפל בקשת שבין הצומת הזה להורה שלו במסלול נקבל אי שיוויון שיראה שהעץ לא אופטימלי.

סבוכיות החלק הראשון היא $O(|V|)$ כי בעץ יש רק $V-1$ קשתות. סבוכות החלק השני היא $O(|V|+|E|)$ כי עוברים על פני כל קשת מספר סופי של פעמים.

נתונה רשת זרימה, $G=(V,E)$, מקור s, בור t, ופונקצית קיבול $c:E\to R^+\cup\{0\}$; כמו-כן נתונה קשת $e=(u,v)$ ב- E. תארו אלגוריתם **יעיל**, המוצא מבין כל הזרימות המקסימליות מ- s ל- t, זרימה מקסימלית f ברשת, עבורה $f(e)$ היא הזרימה המקסימלית האפשרית על הקשת e.

הסבירו את נכונות האלגוריתם וחשבו את סבוכיותו.

סבוכיות של אלגוריתם לחשוב זרימה מקסימלית למשל דיניץ: $O(|V|^2|E|)$

תאור האלגוריתם

שלב ראשון: נחשב זרימה מקסימלית מ- s ל- t.

שלב שני: לאחר שהתקבלה הזרימה המקסימלית בשלב הראשון, נחשב רשת חדשה. נוסיף לצמתים המקוריים שני צמתים u_1 ו v_1. יהיה אוסף חדש של קשתות. תהיה קשת מ- v_1 ל- v ותהיה קשת מ- u ל- u_1. הקיבול בכל אחת מקשתות אלה יהיה שווה למקסימום הזרימה שאפשר להוסיף בקשת (u,v) ביחס לזרימה של השלב הראשון, כך למשל אם החסם העליון הוא 8 יחידות ובשלב הראשון הזרמנו 5 יחידות אז לקשתות החדשות יהיה קיבול 3. את הקשת המקורית בין u ל v ננתק. עבור כל זוג צמתים אחרים שמחוברים בקשת ברשת המקורית, נקבע קשתות בכל אחד משני הכיוונים שקיבולן יהיה שווה למקסימום הזרימה שאפשר להוסיף בכיוון זה ביחס לזרימה של סעיף א', כך למשל אם בקשת ניתן להזרים עד 9 יחידות וכעת מוזרמים בה 7 יחידות, אז בכיוון אחד תהיה קשת בעלת קיבול 7 וקשת שניה בעל קיבול 2. כעת נחשב זרימה מקסימלית מ- v_1 ל- u_1.

הזרימה המבוקשת הסופית על כל קשת תהיה שווה לסכום הזרימה עליה בשלב הראשון והזרימה בין שני קצוותיה בשלב השני. הזרימה על הקשת e תהיה שווה לסכום הזרימה עליה בשלב הראשון והזרימה מ- v_1 ל- v בשלב השני.

הסבר נכונות

בשלב השני אנו משמרים את הזרימה על כל הצמתים מבלי לחרוג מהקיבולים המותרים, כך שנשמרת הזרימה המקסימלית מ- s ל- t. אך במסגרת זו מוגדלת ככל האפשר הזרימה על הקשת e. אי אפשר להשיג זרימה יותר גדולה מ- s ל- t מאשר זאת שהושגה בשלב הראשון. על הצמתים שהם לא u ולא

v חייבים לשמר את המאזן בשלב השני, כך שמה שנכנס אליהם גם יוצא מהם. כמוכן, כדי להזרים יותר מאשר בשלב הראשון מ- u ל- v, צריך שיכנס ל- u יותר וצריך שיצא מ- v. מה שנכניס יותר ל- u זה מה שיזרום בקשתות החדשות בשלב השני.

יהא $G=(V,E)$ גרף מכוון עם משקלות חיוביים על הקשתות, המיצג ע"י רשימות שכנות. הגרף מייצג מפת כבישים, כשמשקל כל קשת הוא אורך הקשת שהיא מייצגת.

בחלק מהצמתים נמצאות תחנות דלק.

תארו אלגוריתם יעיל ככל האפשר המוצא לכל צומת בגרף תחנת דלק הכי קרובה ואת המרחק אליה.

יעילות $O(|E|+|V|\log|V|)$

אלגוריתם נוסיף צומת s שיהיו לו רק קשתות שיוצאות לצמתים שמייצגים תחנות דלק. נהפוך את הכיוון של יתר הקשתות. נריץ את האלגוריתם של דיקסטרה למציאת מסלולים לכל הצמתים מצומת s. עבור כל צומת, המשקל שימצא אליו הוא המרחק ממנו לתחנת הדלק הקרובה ביותר אליו. בעץ שהתקבל, נמצא עבור כל צומת שמייצג תחנת דלק, את קבוצת הצמתים שניתן להגיע אליה ממנו. לגבי כל צומת, תחנת הדלק הכי קרובה היא התחנה שמייצג הצומת שממנו הגענו אליו.

הסבר - מסלול מצומת אל תחנת דלק הוא מסלול מתחנת דלק לאותו צומת על הגרף שכיוון קשתותיו הפוך. לגבי כל צומת, המסלול לצומת s עובר בתחנת הדלק הקרובה ביותר אליו. האלגוריתם של דיקסטרה יוצר עץ מסלולים ולכן לגבי כל צומת, יש מסלול רק מתחנת דלק אחת.

נתון גרף מכוון וקשיר $G=(V,E)$ עם משקלות חיוביים על הקשתות, המיוצג ע"י רשימות שכנות.

לכל $t>0$ נגדיר את $E(t)$ להיות קבוצת הקשתות שמשקלן הוא לכל היותר t. תארו אלגוריתם יעיל ככל האפשר המחשב את t המינימלי עבורו $G(t)=(V,E(t))$ הוא קשיר.

יעילות $O(|E|+|V|\log|V|)$

אלגוריתם - נבחר צומת שרירותי. נסמן אותו ככזה שהגענו אליו. נסמן את יתר הצמתים ככאלה שלא הגענו אליהם עדיין. בדומה למה שמבוצע באלגוריתם של פרים, נצרף בכל שלב לקבוצת הצמתים שהגענו אליה צומת אחד. זה יהיה הצומת שלו קשת בעלת משקל מינימלי לצומת שכבר הגענו אליו. נמשיך בתהליך עד שנגיע לכל הצמתים. כעת נהפוך את כיוונן של כל קשתות הגרף ושוב נתחיל באותו צומת שרירותי ובתהליך דומה נוסיף צמתים עד שנגיע לכל צמתי הגרף. t יהיה שווה למכסימום בין משקליהן של הקשתות הכבדות ביותר שנמצאו בשני התהליכים.

הסבר - אם יש מצומת מסוים מסלול לכל צומת אחר ומכל צומת יש מסלול אליו, אז יש דרכו מסלול מכל צומת לכל צומת אחר. מסלול מצומת אחר אליו משתמש באותן קשתות כמו מסלול ממנו לאותו צומת אחר בגרף שכיוון קשתותיו הפוך. בכל שלב אנו בוחרים בקשת בעלת משקל מינימלי שמאפשרת קשירות בין שתי קבוצות של צמתים.

נתון גרף מכוון G=(V, E) המיוצג ע"י רשימות שכנות עם פונקצית משקל חיובית w : E → R$^+$ ונתון s∈V . בנוסף, נתונה פונקציה f : V → R. תארו אלגוריתם יעיל ככל האפשר הבודק האם לכל v∈V מתקיים δ(s,v)=f(v), כאשר δ(s,v) מסמן את המרחק הקצר ביותר מ-s ל-v בגרף.

יעילות: $O(|V|+|E|)$

אלגוריתם: -

נעבור על כל הקשתות. עבור כל קשת $e_{i,j}$ נוודא ש $f(v_j)\le f(v_i)+w_{i,j}$. כמו-כן נוודא שעבור כל צומת j קיימת קשת $e_{i,j}$ כך ש $f(v_j)=f(v_i)+w_{i,j}$. אם"ם כל אלה יתקיימו אז אלה המרחקים המינימלים מצומת s.

הסבר:

אם קיים צומת k שעבורו $f(v_k)>\delta(s,v_k)$ אז במסלול הקצר ביותר לצומת v_k קיים צומת ראשון v_j שעבורו $f(v_j)>\delta(s,v_j)$. אם v_i הוא אביו של v_j במסלול אז דרכו נקבל ש $f(v_j)>f(v_i)+w_{i,j}$.

נניח שקיים צומת k כך שעבורו $f(v_k)<\delta(s,v_k)$: נסתכל על הצומת j שעבורו $f(v_j)$ מינימלי מבין הצמתים שלגביהם $f(v_r)-\delta(s,v_r)$ מקבל ערך מינימלי. עבור צומת זה לא קיים אף צומת i כך ש $f(v_j)=f(v_i)+w_{i,j}$ הסבר: עבור כל צומת i מתקיים לפחות אחד מהשניים $f(v_i)\ge f(v_j)$

או

$f(v_j)-f(v_i)<\delta(s,v_j)-\delta(s,v_i)\le w_{i,j}$.

אם הפונקציה f היא פונקצית מרחקים אז היא עונה לכל האילוצים. לכל צומת מתקבל מרחק מינימלי דרך אביו בעץ המסלולים המינימלים. על כל צומת ועל כל קשת אנו עוברים מספר קבוע של פעמים.

נתונה תבנית P=P[1]P[2]...P[m] מעל א"ב סופי {*}∪Σ שמכילה בדיוק הופעה אחת של הסימן * (למשל: ac*ba). כמו כן נתון טקסט T=T[1]T[2]...T[n] מעל הא"ב Σ. תארו אלגוריתם יעיל ככל האפשר שיקבע האם T מכיל מחרוזת כלשהיא שמתקבלת מ-P ע"י הצבת תת מחרוזת כלשהיא במקום * (למשל: הטקסט d**ac**da**ba**bd מכיל את התבנית ac*ba החל מהאות השניה).

יעילות : $O(n+m)$

אלגוריתם והסבר:

באמצעות אלגוריתם KMP נחפש את ההופעה הראשונה של תת המחרוזת שלפני * ונחפש את ההופעה המאוחרת ביותר של תת המחרוזת שאחרי *. נדרוש שהשניה תתחיל לאחר שהאחרונה תסתיים. אם * צריך להופיע במקום מחרוזת לא ריקה אז נדרוש שיהיה פער בין מקום הסיום של הראשונה ומקום ההתחלה של השניה. אם זה מתקיים אז נוכל להציב את * במקום מה שמפריד את הופעתם.

שני שחקנים נבונים, A ו-B, משחקים במשחק הבא: נתונה סדרה של *n* מספרים שלמים מסודרים בשורה משמאל לימין. השחקנים משחקים לסירוגין כשכל אחד בתורו לוקח את המספר הימני ביותר או את המספר השמאלי ביותר.השחקן A משחק ראשון. בסיום המשחק, הערך עבור השחקן A הוא סכום המספרים שהוא בחר פחות סכום המספרים ש-B בחר. הערך עבור השחקן B הוא, באופן דומה, סכום המספרים שהוא בחר פחות סכום המספרים ש-A בחר. מטרת כל שחקן להשיג ערך מירבי; לכן, למשל, אם הסדרה ההתחלתית 1,3 והשחקנים משחקים אופטימלית אזי הערך עבור A הוא 2 והערך עבור B הוא 2-, ואם הסדרה ההתחלתית היא 1, 8, 3 אזי הערך עבור A הוא 4- והערך עבור B הוא 4.

תארו אלגוריתם יעיל ככל האפשר שיחשב, בהינתן סדרה באורך *n* את ערך המשחק עבור השחקן A.

יעילות: $O(n^2)$

אלגוריתם והסבר: יהיו $a_1,a_2,\cdots,a_n$ איברי הסדרה. נגדיר $f_{i,i}=a_i$ עבור כל $1\le i\le n$. עבור כל $1\le i<j\le n$ נגדיר $f_{i,j}=\max\{a_i-f_{i+1,j},a_j-f_{i,j-1}\}$. שחקן יכול לקחת מאחד הקצוות שנותרו ואז התור עובר לאחר. עבור כל i,j : $1\le i<j\le n$ יש השוואה בין שני גדלים.

נתון טקסט $T=T[1]T[2]\cdots T[n]$ ומחרוזת $P=P[1]P[2]\cdots P[m]$ מעל א"ב סופי $\sum$, ונניח כי כל התווים במחרוזת P שונים. תארו אלגוריתם יעיל ככל האפשר שימצא את כל ההסטים s , $0\le s\le n-m$, עבורם P מופיעה בהסט s ב- T עם לכל

היותר 3 שיבושים, ז"א $\left|\left\{i \middle| 1 \le i \le m, P[i] \ne T[s+i]\right\}\right| \le 3$.

סיבוכיות: $O(n+m)$

הסבר: - נתחיל להשוות בין הטקסט למחרוזת החל מההתחלה עד מקום $k+1$ בו נגלה אי התאמה רביעית או התאמה מספקת (עם לכל היותר 3 שגיאות)לכל המחרוזת. אם $k<7$ אז נתחיל תהליך זהה במקום אחד יותר מאוחר מהמקום שבו התחלנו פעם קודמת. אם $k \ge 7$ אז נתחיל ב $k-6$ מקומות אחרי המקום שהתחלנו פעם קודמת.

לא תתכן חפיפה של יותר מ 6 תווים בין שתי רישות שונות ששתיהן מדויקות מספיק. בשתי רישות שמתאימות מספיק יש לכל היותר $2\cdot 3=6$ מקומות שבהן התו בטקסט לא שווה לתו באחת הרישות. ביתר המקומות הוא צריך להיות שווה לתווים בשתי הרישות. אך כל התווים במחרוזת שונים והוא לא יכול להיות שווה לשני תווים שונים.

אנו משקיעים $k+1$ השוואות כדי להתקדם $k-6$ צעדים או לכל היותר 6 השוואות כדי להתקדם צעד אחד. לכן הסיבוכיות היא לינארית.

נתונה מטריצה n על n שאיבריה הם 0 או 1. אלכסון מוכלל של המטריצה הוא של n אחדות כך שמכל שורה ומכל עמודה נבחר אחד יחיד. לדוגמא , במטריצה

0	1^*	0	1
0	0	1^*	1
1^*	0	1	0
0	1	0	1^*

משמאל מודגש אלכסון מוכלל. שימו לב שלא בהכרח קיים אלכסון מוכלל למטריצה.

להלן אלגוריתם לחישוב אלכסון מוכלל בהינתן המטריצה:

1. נחזיק מערך בוליאני בגודל n שבו נסמן כל טור שכבר "תפסנו" (בחרנו בו כבר '1 '). בהתחלה כל

הטורים אינם תפוסים.

2. נעבור על המטריצה שורה-שורה:

א. בכל שורה נחפש את ה-1 הראשון בטור שעוד לא תפסנו.

ב. אם נמצא 1 כזה, נוסיף אותו לאלכסון המוכלל ונסמן שהטור תפוס.

ג. אם לא נמצא 1 כזה בשורה הנוכחית-נעצור ונודיע שאין פתרון.

א) הוכיחו כי האלגוריתם שגוי.

נראה דוגמא שבה האלגוריתם לא נותן פתרון, למרות שיש פתרון.

אם נבחר את המשבצת השמאלית העליונה אז כבר לא יהיה פתרון. אבל יש פתרון כאשר נבחרת המשבצת האמצעית העליונה. אפשר גם לתת דוגמא עם מטריצה 2×2 .

ב) תארו אלגוריתם יעיל ככל האפשר שימצא אלכסון מוכלל של המטריצה, אם קיים, או יאמר בביטחון

שאין.

יעילות: של חישוב זיווג מכסימלי בגרף דו צדדי

אלגוריתם והסבר: נגדיר גרף דו צדדי שבו הצמתים מצד אחד ייצגו את השורות והצמתים מהצד השני ייצגו את העמודות. תהיה קשת מצומת i לצומת j אם"ם במטריצה במשבצת ה- i, j מופיע 1.

בגרף זה נחשב זיווג מכסימלי. קיים פתרון אם"ם בגרף זה קיים זיווג מושלם. בזיווג מושלם, בכל צומת נוגעת קשת אחת. את בעיית הזיווג אפשר לפתור באמצעות פתרון בעיית זרימה. נוסיף מקור שממנו יצאו קשתות בקיבול 1 כל אחת לכל צמתי צד אחד ונוסיף יעד כך שמכל אחד מצמתי הצד השני יצאו קשתות אל היעד. לכל הקשתות יהיה קיבול 1 בכיוון מהצד הראשון לשני. ננסה להזרים n יחידות.

נתון גרף אציקלי מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות ונתון זוג צמתים $x, y \in V$. תארו אלגוריתם יעיל ככל האפשר שיחשב את מספר המסלולים המכוונים בגרף G העוברים גם דרך x וגם דרך y .

יעילות: $O(|V|+|E|)$

אלגוריתם והסבר: - נחשב מיון טופולוגי בגרף. נניח בלי הגבלת הכלליות שבמיון מופיע x לפני y .

1	1	0
1	0	0
0	0	1

נוסיף צומת s ונחברו בקשתות לכל הצמתים שמופיעים במיון הטופולוגי לא אחרי x . נוסיף צומת t ונחבר קשתות אליו מכל הצמתים שמופיעים במיון הטופולוגי לא לפני y .

נחשב את מספר המסלולים מ s ל x , מ x ל y ומ y ל t. הפתרון יהיה המכפלה של שלושת המספרים האלה. בחישוב כל אחד ממספרים אלה, נסתמך על המיון הטופולוגי ונשתמש בתכנות דינמי.

למשל בחישוב מספר המסלולים מ x ל y , לגבי צומת w שבינהם כולל אותם, מספר המסלולים מ w ל y יהיה שווה לסכום מספרי המסלולים ל y מצמתים שיש מ w קשת אליהם (ברור שמסלול חייב לעבור דרך אחד מהם או שהוא ישיר ל y - שזהו מקרה פרטי).

נתון גרף לא מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות ונתונה פונקצית משקל $w: E \to R$.

ידוע כי $|E|=|V|+10$. תארו אלגוריתם יעיל ככל האפשר שימצא עץ פורש מינימלי ב- G.

יעילות: $O(|V|+|E|)$

אלגוריתם והסבר: 1. נחשב את הרכיבים הדו קשירים בגרף.

2. בכל רכיב דו קשיר נמצא את הקשת הכי כבדה ונמחק אותה.

כל עוד נשארו לפחות $|V|$ קשתות- נחזור על שלבים 1 ו 2.

גשרים חייבים להיות בעץ. קשת שהיא הכבדה ביותר ברכיב דו קשיר היא על מעגל שבו כל הקשתות לא כבדות ממנה. לכן למשל לפי הנכונות של האלגוריתם של קרוסקל יש עץ שהיא לא כלולה בו.

מכיון שבתחילה $|E|=|V|+10$ אז נחזור על התהליך לא יותר מ 10 פעמים.

יהא $G=(V,E)$ גרף לא מכוון מיוצג ע"י רשימות שכנות, עם משקל שלם $w(e)$ לכל קשת $e\in E$. לכל מספר ממשי x, נסמן ב- $G^{(x)}=\left(V,E^{(x)}\right)$ את תת הגרף המורכב מכל הקשתות שמשקלן לכל היותר x. תארו אלגוריתם יעיל ככל האפשר המחשב את ה- x המקסימלי עבורו $G^{(x)}$ חסר מעגלים.

יעילות:
$$O\left(\min\left(|V|,|E|\right)\log\left(|E|\right)+|V|+|E|\right)$$

אלגוריתם: נזרוק מהגרף את כל הקשתות שהן לא בין $|V|$ הקשתות הקלות ביותר.

נבצע את השלב הראשון של האלגוריתם של קרוסקל (מיון הקשתות לפי משקלן).

נבצע את השלב השני ונקטע אותו כאשר תמצא קשת שסוגרת מעגל עם קשתות שלא כבדות ממנה.

עבור כל מספר x שקטן ממש ממשקל קשת זו, $G^{(x)}$ הוא חסר מעגלים.

הסבר: כל עוד לא התגלתה קשת שסוגרת מעגל אז אפשר לבחור את כל הקשתות מבלי שיווצר מעגל.

בכל קבוצה של $|V|$ קשתות יש בודאי קשת שיוצרת מעגל עם קשתות שאינן כבדות ממנה.

אפשר למצוא את הקשתות שצריך למחוק בזמן לינארי (סטטיסטי סדר).

נתון גרף מכוון $G=(V,E)$ על ידי רשימות שכנויות, ומשקל $w(e)$ חיובי לכל $e\in E$. הגרף מייצג רשת תקשורת, כאשר כל צומת מייצג משתמש, וכל קשת מייצגת קשר בין שני משתמשים. משתמש u יכול לשלוח הודעה למשתמש v אם קיים מסלול (מכוון) מ- u ל- v בגרף. משקל הקשת $(u,v)\in E$ הוא עלות הניתוק של הקשר הישיר בין המשתמשים u ו- v. כל צומת $v\in V$ מסווג כ"בוגר", "קטין", או "לא ידוע".

מטרתנו היא לחשב תת-קבוצה E' של קשתות הגרף כך שאם נמחק את כל קשתות E' מהגרף, לא יהיה מסלול משום צומת "קטין" לשום צומת "בוגר". תארו אלגוריתם יעיל ככל האפשר המקבל כקלט את w,G, ואת סיווג הצמתים, ומחזיר קבוצת קשתות E' המקיימת את התכונה לעיל, בעלת משקל כולל מינימלי מבין כל הקבוצות המקיימות את התכונה.

יעילות: של אלגוריתם לחישוב זרימה מקסימלית

אלגוריתם: נוסיף מקור ויעד. נוסיף קשתות מהמקור לכל צומת "קטין" ומכל צומת "בוגר" ליעד. נתן קיבולים לכל קשתות הגרף. הקיבול של כל קשת מקורית יהיה שווה למשקלה והקיבול של כל קשת שהוספה יהיה שווה לסכום כל קיבולי הקשתות המקוריות. נפעיל אלגוריתם לחישוב זרימה מקסימלית מהמקור ליעד.

אלגוריתם זה ימצא חתך מינימלי. הקשתות שנחזיר הן קשתות החתך.

הסבר: יש להסיר את כל הקשתות באיזשהו חתך כדי לנתק את הקשר. מכיון שלקשתות החדשות נתנו קיבול גדול אז החתך יהיה בין צמתים מהגרף המקורי.

הערה: את הקשתות המחברות "קטין" ל"בוגר" יכולנו מראש להסיר. יש גרפים בהם זה משנה את סדר הגודל של היעילות (כאשר מספר הקשתות הנוגעות ב"לא ידועים" הוא קטן).

נתונה קבוצה E של n האינטרוולים על הישר $[a_1,b_1],\cdots,[a_n,b_n]$ כשלכל אינטרוול $[a_i,b_i]$ יש משקל חיובי w_i. תארו אלגוריתם יעיל ככל האפשר שמחשב תת-קבוצה E' של E עם משקל מקסימלי כך שכל האינטרוולים ב- E' זרים בזוגות (החיתוך של כל שניים ריק). (**רמז**: סדרו את האינטרוולים בסדר עולה של נקודות הקצה הימניות שלהם, וקבלו נוסחת נסיגה עבור $OPT(j)$, האופטימום עבור j האינטרוולים הראשונים.)

יעילות: $O(n\ln(n))$

אלגוריתם:

שלב ראשון: נמיין את האינטרוולים לפי סדר עולה של הקצה הימני שלהם.

שלב שני: עבור כל אינטרוול j נמצא באמצאות חיפוש בינארי מבין האינטרוולים שקדמו לו במיון את האינטרוול $m(j)$ בעל הקצה הימני ביותר שאיתו הוא לא נחתך.

שלב שלישי: נעבור על פני כל האינטרוולים על פי הסדר שנקבע בשלב הראשון. עבור כל j נגדיר

$$T(j)=Max\{OPT(j-1),OPT(m(j))+w_j\}$$

$$OPT(0)=0$$

הסבר: בכל שלב אנו מקבלים את הפתרון הטוב ביותר שלא כולל את האינטרוול הנבדק או שכולל אותו וגם קבוצה מקסימלית מבין האינטרוולים הקודמים שאינם נחתכים איתו.

נתון גרף מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות, ונתונה תת-קבוצה $E^{'}\subseteq E$ של קשתות "חשובות".

תארו אלגוריתם יעיל ככל האפשר, הבודק אם קיים ב- G מסלול העובר דרך כל הקשתות של $E^{'}$.

<u>**יעילות**</u>: $O(|V|+|E|)$

<u>**אלגוריתם**</u>: בגרף הנתון נוסיף עבור כל קשת חשובה צומת ונפצל את הקשת לשתי קשתות. הראשונה תחבר את צומת המקור לצומת הנוסף של אותה קשת והשניה את הצומת הנוסף לצומת היעד של הקשת. נריץ את האלגוריתם שתואר בתשובה לשאלה 21 בקובץ זה לבדיקה האם הצמתים שהוספו נמצאים על מסלול אחד

<u>**הסבר**</u>: יש מסלול העובר בכל הצמתים שהוספו אם"ם יש בגרף המקורי מסלול שעובר בכל הקשתות המיוחדות.

לא הגדלנו את סדר הגודל של סכום מספר הצמתים והקשתות בגרף.

<u>הערה</u>: אם נדרוש שהמסלול יהיה פשוט אז הבעיה היא NPC.

נתונה סדרה $X=x_1x_2\cdots x_n$ כאשר לכל $1\le i\le n$, $x_i\in\{a,b,c\}$. תאר/י אלגוריתם יעיל ככל האפשר המחשב רצף $x_ix_{i+1}\cdots x_{j-1}x_j$ עבורו מספר ההופעות של a פחות מספר ההופעות של b מקסימלי. במילים אחרות,

$$\left|\{p:i\le p\le j,x_p=a\}\right|-\left|\{p:i\le p\le j,x_p=b\}\right|$$

מקסימלי מבין הבחירות האפשריות ל- i,j עבור $1\le i\le j\le n$.

<u>**יעילות**</u>: $O(n)$

<u>**אלגוריתם והסבר**</u>:

בעזרת תכנות דינמי נחשב את ערכי F_p שמייצגים את הפתרון הטוב ביותר שמתחיל בנקודה p.

$$F_n=\begin{cases}0 & x_n\in\{b,c\}\\ 1 & x_n=a\end{cases}$$

ועבור כל $1\le p\le n-1$:

$$F_p=\begin{cases}F_{p+1}+1 & x_p=a\\ F_{p+1} & x_p=c\\ \max\{F_{p+1}-1,0\} & x_p=b\end{cases}$$

אחר-כך נעבור על פני כל ערכי F_p. הרצף המבוקש יתחיל בנקודה שבה F_p הוא מקסימלי ויסתיים בנקודה המוקדמת ביותר אחר-כך שבה הוא אפס, או ימשך עד הסוף אם עד הסוף אין נקודה שבה הפונקציה מתאפסת. אם מנקודה מסוימת אי אפשר ליצור רצף שבו יותר a מ b, אז כדאי לשמוט את הקרן שמתחילה בנקודה זו. אחרת כדאי לצרף את הרצף הזה לפתרון שמגיע עד אליה.

נתונה רשת זרימה $G=(V,E)$ עם מקור s, בור t ופונ' קיבול $c:E\to R^{+}$ ונתונה זרימה f ברשת.

נניח שמצאנו מסילה משפרת $P:s\rightsquigarrow t$ ברשת השיורית G_f שאורכה בקשתות מינימלי. תהיה g הזרימה ברשת G המתקבלת מ- f לאחר שהזרמנו לאורך P את הקיבול השיורי $c(P)$.

הוכח/הוכיחי כי לכל צומת $v\in V$ מתקיים $\delta_g(s,v)\ge\delta_f(s,v)$ כאשר δ_f,δ_g הם המרחקים ברשתות השיוריות G_f,G_g בהתאמה.

<u>**הוכחה**</u>: נניח שיש צמתים שעבורם זה לא מתקיים. אז לאחר העדכון קיים צומת v שהוא הראשון במסלול שמוביל אליו שעבורו המסלול החדש קצר יותר. לאחר העדכון מגיעים אליו מצומת u שאליו כעת המסלול קצר יותר ב 1 ולא יותר קצר מאשר היה קודם. קודם לא יכולנו להגיע ל v דרך u (כי אחרת המסלול ל v לא היה ארוך יותר קודם). לכן נוכל להסיק שבעדכון הזרמנו מ v ל u (כך שהתאפשרה אחר-כך זרימה בכיוון ההפוך). מכיון שהמרחק ל v היה קודם ארוך יותר מאשר עכשיו ומ v עברנו ל u אז המרחק ל u היה קודם ארוך יותר מאשר עכשיו ל v ובודאי גם מהמרחק עכשיו ל u.

זאת סתירה לכך שקיימים צמתים שהמרחק אליהם התקצר.

נתון גרף $G=(V,E)$ מכוון וקשיר בחוזקה המיוצג ע"י רשימות שכנות, נתונה פונקצית משקל $w:E\to R$ ונתונים זוג צמתים שונים $u,v\in V$. ידוע שאין בגרף מעגל שלילי ביחס ל-w.

תאר/י אלגוריתם יעיל ככל האפשר שימצא, עבור כל ערך של k, $2\le k\le|V|-1$, את המשקל הקל ביותר של מסילה מכוונת $u\rightsquigarrow v$ שאורכה לכל היותר k קשתות (אם יש כזו).

הערה: האלגוריתם מחשב, אם כן, $|V|-2$ ערכים מ- $R\cup\{\infty\}$.

<u>**יעילות**</u>: $O(|V||E|)$

<u>**אלגוריתם והסבר**</u>:

נבנה גרף בעל $|V|$ שכבות של צמתים.

בשכבה הראשונה יהיה רק הצומת u. בכל אחת מהשכבות האחרות יהיה עותק של כל צומת מצמתי הגרף. מהשכבה הראשונה לשכבה השניה יצאו כל הקשתות שיוצאות מהצומת u בגרף המקורי. מצומת שבשכבה אחרת תצא קשת לנציג של צומת אחר שבשכבה הבאה אם"ם יש ביניהם קשת בגרף המקורי. כמו-כן תצא קשת מכל נציג לנציג של אותו צומת שבשכבה הבאה. משקלי הקשתות שבין נציגים של צמתים שונים יהיו תמיד זהים למשקלם בגרף המקורי ומשקלי הקשתות שבין נציגים של

אותו צומת יהיו תמיד שווים לאפס. בהינתן המרחק לצמתים שבשכבה מסוימת, יהיה המרחק לכל צומת שבשכבה הבאה שווה למינימום על-פני הצמתים שבשכבה הקודמת, שמהם נכנסת לצומת קשת, של המשקל אליהם ועוד משקל הקשת לצומת שבשכבה הבאה. המשקלים המבוקשים יהיו המשקלים לנציגי הצומת v שבשכבות השונות. בהינתן שחישבנו את המשקלים לשכבה קודמת אז המשקלים לשכבה הבאה יכולים להשתמש בקשת אחת יותר או לא להשתמש ביותר קשתות ואז משתמשים בקשת בעלת משקל של אפס. אם עד שכבה מסוימת אין מסלול אז המשקל של מסלולים בעלי אורך לכל היותר זה הוא ∞. מכיון שהגרף הוא קשיר בחוזקה אז מספר הקשתות בו אינו קטן ממספר צמתיו. עבור כל זוג שכבות שכנות יהיה לנו סדר גודל של $|E|$ השוואות.

בבית הספר היסודי "סדר מעל לכל" לומדים תלמידים בכיתות א'-ו'. בכל כיתה לומדים n תלמידים.

במסדר הסיום רוצה המנהל לסדר את התלמידים ב־n טורים כשבראש כל טור ילד מכיתה א', אחריו ילד מכיתה ב', בעקבותיו ילד מכיתה ג', וכך הלאה עד לסוף הטור שבו ילד מכיתה ו'. לכל ילד מכיתה ב' יש רשימה של ילדים מכיתה א' שהוא מוכן לעמוד בעקבותיהם בטור. באופן דומה, לכל ילד מכיתה ג' יש רשימה של ילדים מכיתה ב' שהוא מוכן לעמוד בעקבותיהם בטור, וכך הלאה עד לילדי כיתה ו' שלכל אחד מהם רשימה של ילדים מכיתה ה' שהוא מוכן לעמוד בעקבותיהם.

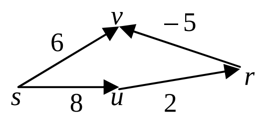
המנהל מטיל על המורה לאלגוריתמים לבדוק האם אפשר לסדר את $6n$ הילדים ב־n טורים תוך שמירה על אוסף האילוצים הנתון, תאר/י אלגוריתם יעיל ככל האפשר בו תשתמש המורה כדי לפתור את הבעיה.

יעילות: של אלגוריתם לחישוב זיווג מכסימלי בגרף דו-צדדי

אלגוריתם והסבר:

נפתור 5 בעיות זיווג. עבור כל i: $2 \le i \le 6$ נחשב זיווג מכסימלי בגרף הדו-צדדי ששתי קבוצות צמתיו מייצגות את ילדי כיתה i ואת ילדי כיתה $i-1$. בין כל זוג צמתים כאלה יש קשת אם"ם הילד מכיתה i מוכן לעמוד מאחורי הילד מכיתה $i-1$. אם"ם בכל 5 הבעיות יש זיווג מושלם אז יש פתרון כנדרש.

סעיף א'



תאר/י דוגמא של גרף $G=(V,E)$ עם פונ' משקל $w:E\to R$ וצומת $s\in V$ כך שאין מעגל שלילי ביחס ל־w בגרף והאלגוריתם של Dijkstra נכשל; במילים אחרות, הראה/הראי צומת $v\in V$ כך שהאלגוריתם אינו מחשב נכון את אורך המסילה הקלה ביותר מ־s ל־v.

דוגמא והסבר:

הקשת מ־s ל־v היא הקשת הקלה ביותר שיוצאת מ־s. לכן כבר בשלב הראשון יקבע מרחק סופי 6 מ־s ל־v. אך דרך u ו r יש מסלול מ־s ל־v שאורכו $8+2-5=5$.

סעיף ב'

נתון גרף לא מכוון $G=(V,E)$ המיוצג ע"י רשימות שכנות עם פונ' משקל חיובית $w:E\to R^+$ ונתון צומת $s\in V$. ידוע שדרגת כל צומת בגרף G היא לכל היותר 10.

תאר/י אלגוריתם יעיל ככל האפשר שימצא את 100 הצמתים הקרובים ביותר ל־s.[1]

[1] במקרה של שיוויון בין מרחקים נעדיף צמתים עם מס' סידורי קטן יותר.

יעילות: $O(1)$

אלגוריתם והסבר:

נריץ את אלגוריתם דיקסטרה עד הוספת 100 צמתים לקבוצת הצמתים שאת המרחק הסופי אליהם חישבנו. בכל שלב נחשב את המרחק לכל השכנים של הצמתים שכבר בקבוצה. מרחק זה יהיה שווה למינימום על פני הסכום של אורכי המסלולים לצומת שכבר בקבוצה + אורך הקשת מהצומת הזה לצומת שעדיין לא בקבוצה. בכל שלב נוסיף לקבוצה את הצומת בעל האינדקס המינימלי מבין הצמתים הקרובים ביותר שעדיין לא היה בקבוצה. המרחק הזה אליו יהיה סופי. בכל שלב לא עוברים בבדיקה על יותר מ $100*10=1000$ קשתות.

הערה: כדי להגיע לסיבוכיות זו, לא נזקקנו למבנה הנתונים שבו משתמשים בדרך כלל באלגוריתם דיקסטרה.

עצים

עצים 1

מהו עץ חופשי ?
גרף בלתי מכוון, קשיר וללא מעגלים

האם לעץ חופשי יש שורש ? למה ?
לא
כל צומת היא השורש באותה מידה ?

האם כל גרף בלתי מכוון, קשיר וללא מעגלים הוא עץ ?
כן

האם ייתכן עץ המכיל פחות מ n-1 קשתות, כאשר n הוא מספר הצמתים ? יותר מ n-1 ? למה ?
לא
תמיד שווה n-1
הסבר אינטואיטיבי : לכל צומת צמודה הקשת שמחוברת אליו מלמעלה, ורק לשורש אין קשת..

מהו יער ?
מספר עצים לא מקושרים

מהו גרף עם מעגל ?
לא עץ ולא יער

מהו עומק השורש ?
0

מהם סוגי הצמתים ?
שורש
עלים
פנימיים

האם בעץ חופשי כל צומת יכול להיות שורש ? למה ?
כן
כי הגדרת הצורה מאפשרת זאת

האם ברגע שנקבע השורש הופך העץ החופשי למכוון ?
בעצם כן, למרות שזו לא הגדרה פורמלית

מהו עץ מלא ?
לכל סדור שבו דרגת כל צומת פנימי היא 2

מהו עלה ?
צומת ללא בנים

מהו עומק ?
אורך המסלול מעלה לשורש

מהו גובה ?
העומק הגדול ביותר של צומת

מהו עץ מושרש ?
שיש לו קודקוד ?

כמה עצים מושרשים שונים ניתן לייצר מעץ חופשי אחד ?
כמספר הצמתים
?

האם השורש הוא צומת פנימי ?
לא ?

מהי רמה ?
קבוצת צמתים שמרחקם מהשורש שווה

עץ מסודר, עץ מאוזן, עץ מיקום

מהו עץ סדור ?
עץ מושרש שבו יש משמעות לסדר צמתי
הבן (שמאל או ימין) ?

מהי משוואת ההפרשים ? מה פתרונה ?

מהו עץ מסודר ?
עץ סדור

מהו עץ מיקום ?
לכל ילד יש תפקיד
כמו עץ בינארי
לא חייב להיות מלא

מהו עץ מיקום ?
כל בן מקבל תוית שהיא מספר חיובי שלם

ע"פ מה נקבעות תויות הצמתים בעץ מיקום
?
שרירותית ?

האם בעץ מיקום חשוב הסדר בתוך הרמה
או גם בין הרמות ?
גם בין הרמות ?

מהי הדוגמא הקלאסית לעץ מיקום ?
כל רמה היא אות.

האם ייתכן עץ מאוזן לא שלם ?
כן. הרמה התחתונה לא חייבת להיות מלאה

כיצד מדלגים אינדקס לבן שמאלי, ימני ואב
?
2i + 1
2i + 2
ערך תחתון של (i-1)/2

מהם הייתרונות והחסרונות של עץ מאוזן על
פני מערך ? עבור אילו פעולות ?
חסרונות –
= לא טוב לעץ שאינו כמעט שלם
= מוגבל לזכרון רציף
ייתרון –
= מהיר, לא צריך מצביעים ולכן חוסך את
מקומם

מהו עקרון ההוכחה לכך הגובהו המזערי של
עץ בעל n צמתים הוא log n ?
מראים את גובהו של עץ כמעט מלא = log n
מסבירים את ההבדל בינו לבין עץ לא כמעט
מלא
מסבירים שעץ שאינו כמעט מלא - גבוה יותר

עץ k ארי ובינארי

האם עץ בינארי שלם הוא מלא ? ולהיפך ?
למה ?
כן
לא
שלם מכיל מלא

האם עץ בינארי הוא עץ סדור שדרגת צמתיו
היא לכל היותר 2 ? למה ?
לא
כי בבינארי משנה ימין \ שמאל לכל בן

מהו עץ k ארי ?
ה k מסמל את מספר הבנים המירבי, כמו
בינארי, קייארי...
עץ מיקום שבו לאף צומת אין בנים שתויתם
גדולה מ k

מהו טווח מספר האיברים בשורה התחתונה
של עץ בינארי מלא ?
בין 2^h
ל 2^(h+1) - 1

מהו עץ מאוזן ?
שגובהו logn

האם ייתכן עץ בינארי שאינו עץ מיקום ?
למה ?
לא ?

איך מיישמים עץ בינארי ?
מערך או רשימה מקושרת.
אינדקסים כמו ספר משמאל לימין
או מספר מצביעים מכל צומת לעת k ארי

כמה צעדים צריך לעשות כדי לעבור על עץ ?
למה ?
2n-2 . בכל קשת עוברים פעמיים

מהי שיטת הקופונים ? מה מטרתה ?
שיטה לחישוב ניתוח לשיעורים ?
כל צומת מקבל 2 קופונים ומעביר אחד
לצומת שמעליו.
כמות הקופונים ליד כל צומת מציין את כמות
הצמתים מתחתיו

מה גובהו של עץ בינארי מלא ?
log n

מה ההבדל בין עץ בינארי שלם למלא ?
בשלם, לכל צומת יש 0 או 2 בנים

PRE POST IN ORDER

מהם סוגי הסיורים בעץ ? איך זוכרים את זה ?
Pre – נוכחי > שמאלי > ימני
Post – שמאלי > ימני > נוכחי
In – שמאלי > נוכחי > ימני
Pre – קודם שורש
In – שורש באמצע
Post - שורש בסוף
תמיד שמאל לפני ימין

איך נראה כל סיור על עץ מלא עם 7 צמתים, כשהאינדקסים לפי רמה משמאל לימין ?
pre : 1>2>4>5>3>6>7
post : 4>5>2>6>7>3>1
in : 4>2>5>1>6>3>7

מהו ייתרונו של כל "סיור" ? מתי נשתמש בכל אחד ?
inorder = נותן רשימה ממוינת מעץ חיפוש
post – חישובים אריתמטים

איך מייצרים ביטוי חשבוני על עץ ?
ב data של כל צומת יש מספר או אופרטור , ורושמים את כל העץ ע"פ שיטת סיור בעץ

איפה נוספים סוגריים בסיור חשבוני ?
בתתי עצים, לפי סדר הפעולות

מהו עץ בן שמאלי אח ימני ?
עץ שבו יש מצביעים בין האחים משמאל לימין

עצי חיפוש

אילו עצי חיפוש קיימים ?
עץ חיפוש בינארי
עץ 2-3-4
עץ אדום שחור

מהם הייתרונות והחסרונות של כל עץ
בהשוואה לעצים האחרים ?

מתי נבחר להשתמש בכל עץ ?

מהי סיבוכיות הפעולות עבור עץ 2-3-4 ?

מהו זמן מעבר ממערך ממוין לעץ חיפוש ?
למה ? מהו האלגוריתם ?
O(n)
חציונים ?
אם עץ חיפוש לא מאוזן – פשוט עץ שרוך או ספגטי
אם לא"ש : חציון שורש, ¾ בימני, ¼ בן שמאלי וכך הלאה.אם רמה אחרונה לא מלאה - > אדומים, עלים NIL שחורים

עץ חיפוש בינארי

מהו עקרון במבנה של עץ חיפוש בינארי ?
כל הבנים השמאליים קטנים יותר מהאבא,
כל הימניים גדולים יותר מהאבא

האם עץ חיפוש בינארי תמיד מלא ?
לא. בד"כ אינו מלא

באילו פעולות תומכים ? מהו זמן כל פעולה ?
חפש, מצא מקס \ מינ', הכנס, הוצא,
successor, predecessor
O(log n)

איך מבצעים מחיקה בעץ חיפוש בינארי,
לצומת עם שני בנים ?
מחליפים בין הצומת לעוקב שלו, ומוחקים
את העלה שהיה העוקב

איך מוצאים עוקב ?
אם ל-X יש בן ימני Y, אז העוקב הוא
המינימום של תת העץ שבראשו Y. אחרת,
מטפסים מ-X למעלה כל עוד עוברים מבן
ימני לאבא שלו. כשמגיעים לצומת שהוא בן
שמאלי – אביו הוא העוקב.

האם תתכן הכנסה לשורש עבור עץ לא ריק ?
לא ?

האם העוקב בהכרח תמיד יהיה עלה ? למה ?
כן, כי הוא מינימום מקומי. זה כל הרעיון

האם אפשר לבחור במקום החלפת העוקב,
בהחלפת העוקב מלמטה ? למה ? למה לא ?
כן, כי חוקיות העץ תשמר
סתם פחות מקובל

האם ייתכן מצב שבו החלפה לטובת הוצאה
תגרום לעליית האיבר המוצא ? למה ?
לא
כי א"כ צריך למחוק אותו

מהי סיבוכיות הפעולות עבור עץ חיפוש
בינארי ?
סיבוכיות הזמן לכל פעולה – h גובה העץ.
סיבוכיות מקום: איחסון האיברים – O(n).
מקום נוסף: O(1).

איך מדפיסים רשימה ממוינת מעץ חיפוש
בנארי ?
הולכים עליה בסדר inorder

כיצד ממשים חיפוש על עץ בינארי ?
בלולאה או ברקורסיה

איך מוצאים איבר עוקב בעץ חיפוש בינארי ?
שווה למינימום בתת העץ הימני, או – לאב
הראשון של x ש x אינו בתת העץ הימני
שלו.

האם ניתן להמיר כל עץ חיפוש בינארי לכל
עץ חיפוש בינארי אחר ? מהו המשמעות ?
כן

מהו מספר הרוטציות הדרוש להמרה בין שני
עצים בינארים ? למה ?

עץ אדום שחור

מהו עץ אדום שחור ?
תכונות בסיסיות כשל עץ חיפוש בינארי
גובה שחור קבוע $2\log(n+1)$: מסלול לכל
העלים מכיל אותו מספר שחורים
לכל צומת אחד משני הצבעים
אין במסלול פשוט רצף של שני אדומים
כל העלים שחורים
אם צומת אדום שני בניו שחורים

בגלל איזו בעיה שלא פותר עץ חיפוש בינארי רגיל, נבנה עץ א"ש ?
חיפוש יותר מהיר. עץ בינארי רגיל יכול להיות בגובה n אם הוא כמו חבל עם קשרים..

מהו גובה עץ א"ש ?
מספר השחורים במסלול פשוט ?

האם בהכנסה חשוב לשמור על גובה העץ ? למה ?
כן, זה כל הרעיון
אלא אם נפתחת רמה חדשה ?

האם ייתכן מצב שבו חייבים להוסיף לגובהו של עץ א"ש ? למה ? איך ?
אם נפתחת רמה חדשה

איך יודעים אם נצטרך סיבוב אחד או שניים של העץ ?
אם הצומת הבעייתי פנימי אז שניים ?

האם משנה אם עושים סיבוב > צביעה > סיבוב או סיבוב > סיבוב > צביעה ?
לא

מהם שלושת המקרים בהכנסה ? מה הפתרון ?
דוד אדום – הורים הופכים שחורים וסבא אדום

איזה צבע בתחילה מקבל צומת חדש ? למה ?
אדום
כך לא משתנה גובה העץ

לאיזה עץ דומה עץ אדום שחור ?
חיפוש בינארי

מהי סיבוכיות סיבוב העץ ? למה ?
$O(1)$
כי ייתכנו עד 3 סיבובים, וכל פעולת סיבוב היא $O(1)$

האם ייתכנו אב ובן מאותו הצבע ? למה ?
כן. רק שחור

האם הקשרים בין הצמתים משתנים בסיבוב ? למה ?
לא ?

האם ייתכן רצף של שני אדומים ? למה ?
לא. זו הגדרת העץ

מהי סיבוכיות ייצור עץ ? למה ?
משרירותי nlogn
מממוין n

סיבוכיות הוצאת איבר מעץ ?
log n

האם צריך להכיר את טכניקת הסיבובים למבחן ?
לא ?

מה משותף לכל הפעולות להכנסה ?
האיבר שנכנס למטה הוא אדום
בסוף כל הפעולות צובעים את כל העלים בשחור

מהו בעצם ייתרונו של עץ א"ש על מערך ממוין ?
הוספת איבר לא דורשת הזזת האיברים שאחרי.
ניהול זכרון

האם ניתן לאחד שני עצי א"ש ע"י שורש משותף, ואז שגרת הוצאת שורש ? למה ?
לא, כי אז צריך שעץ אחד יהיה גדול כולו מהשני ?

מהי סיבוכיות איחוד שני עצי א"ש שערכיהם חופפים ? למה ?
$O(n)$
תעביר כל עץ למערך בעזרת ריצה inorder.
תבנה מערך חדש ותכניס את האיברים כך שתשווה כל פעם בין הערכים ותכניס את הקטן מביניהם, שכחתי איך קוראים למיון הזה.
עץ א"ש הנבנה ממערך ממוין = O של N

מה זה הכנה ל LCA ?
Least common ancestor
אב קדמון משותף הנמוך ביותר

מהן ההעצמות השכיחות לעץ א"ש ?
ערכי מיקום
ערך כללי לתתי העצים של כל צומת מצביעים

עץ 4 - 3 – 2 B

לאיזה עץ דומה בעקרון עץ 2 3 4 ?
עץ א"ש ?

מה ההבדל בין עץ B לעץ 2 3 4 ?
אין הבדל ?

עץ TRIE

מה זה TRIE ?

מה מקור הביטוי ?
מתוך retrieval

מה השימושים ?
T9 בסלולרים
מילונים

מהו סדר גודל של חיפוש האות הבאה בכל
צומת ?
תלוי בשיטת אחסון הקשתות הבאות

מהו עץ TRIE ?
עץ שבו לכל צומת 26 בנים שכל אחד מייצג
אות.
וכך כל מילה מיוצגת ע"י מסלול יורד

איך ממשים פונ' זיהוי מילים T9 בפלאפון ?
עץ TRIE

עץ KD

מה מטרתו של העץ ?
עוד סוג של מבנה \ מאגר נתונים שמאפשר
לשמור מידע ולמצא אותו ביעילות יחסית

מה העקרון לבניית העץ ? מהי משוואת
הרקורסיה ?
קוים חוצים אזורים
O(n) = O (n) + 2T(n/2)

מהי סיבוכיות הבנייה ? למה ?
O(nlogn)

האם מיון מוקדם יוסיף לסיבוכיות הבנייה ?
למה ? מה יעזור לבצע מיון מוקדם ?
לא, כי המיון הוא אותה סיבוכיות כמו הבנייה

מהו האזור הקלאסי שנשתמש בעץ KD
לחיפוש ? מה תהיה הסיבוכיות ? למה ?
מלבן
O((sqrt n) + k)

מה הקוד להוספת נקודה ?
צריך לבנות את כל העץ מחדש

ערימות

מהי ערימה ?
כל אב חביב שווה לבניו

האם בערימת מקס', האיבר השני בגודלו
הוא בן של השורש ?
כן

האם בערימת מקס', האיבר הקטן ביותר
הוא עלה ?
כן

מהו תור עדיפויות ? איך עוד קוראים לו ?
ערימה
Heap

מהי סיבוכיות build_max_heap(A) ?
מה עקרון ההוכחה ?
O(n)
אלגוריתם הבניה =
"זריקה" אקראית לתוך ערימה
פעולת heapify = מלמטה למעלה.

$$n\sum_{0}^{\log n}\frac{h}{2^h}=O(n)$$

איפה נעשה שימוש בערימות ?
מערכות הפעלה
C++
שימושים רבים מאוד

מהו זמן heapify ?
O(n)

מהו תהליך ה heapify ?
פעפוע של איבר בודד מטה

איך נקראת הפיכת עץ בינארי לערימה
בינארית ?
build heap

איך עובד אלגוריתם בנית ערימה מעץ ?
מירוץ השוואות מלמטה למעלה

ערימה בינארית

מהי ערימה בינארית ?
עץ בינארי מלא המיושם במערך.
עץ בינארי שמתמלא שורה שורה משמאל לימין.
כל צומת גדול שווה מבניו

האם בערימה בינארית כל בן שמאלי קטן שווה לימני ?
לא

איך מיושמת ערימה בינארית במחשב ?
ע"י אינדקסים של מערך

האם משתמשים במבנה struct כמו בעץ רקורסיבי ? למה ?
לא
פחות יעיל

מהי הפעולה בסיסית בערימה בינארית ?
מהי פעולת ההשוואה הבסיסית לערימה בינארית :
השווה ילדים
השווה מנצח להורה
החלף במידת הצורך
= השוואה משולשת בין ההורה ושני בניו

האם מסדרים מלמטה למעלה או מלמעלה למטה ?
ארגון ערימה שלימה – מלמעלה ?
הוספת איבר אחד – מלמטה
?

למה הסידור הראשוני הוא מלמעלה ?
האם לא "נתקעים" איברים גדולים למטה ?

כמה צמתים בערימה בגובה h ?
2^h

מהי חביבות ?
עצימות בעמידה בתכונה מוגדרת כלשהי
למה נראה שערימה מסודרת אינה ממוינת בשורות התחתונות ?
הסדר בשורה באמת לא משנה

מהן הבעיות ב "הוצא חביב" מערימה בינארית ? מה הפתרון ?
מבנה הערימה נהרס
שמים את האיבר האחרון במקום הראשון ומסדרים שוב

מהו גובהה של ערימה בינארית בת n איברים ? למה ?
Log n
כמו של עץ בינארי מלא

מהי סיבוכיות כל הפעולות בערימה בינארית ?
מצא חביב O(1)
הוצא חביב O(log n)
הוסף איבר O(log n)
בנה ערימה O(n)

Thorax ?
בית החזה

האם ערימה בינארית תמיד מיישמים במערך ? למה ?
כן
בהגדרתה
מערך יותר יעיל מבחינת זכרון כי אין לו צורך בפוינטרים

האם מערך ששומר על תכונת הערימה הוא ממוין ? למה ?
לא, כי האינדקסים שלו "מבולבלים".
וגם כי הערימה עצמה אינה ממוינת

Sift?
לחלחל ?
סנן, נפה, בזק, פזר, בחן
הכוונה בעצם ל heapify
אבל יש גם sift_up וגם sift_down

איך עושים "הוצא חביב" בערימה בינארי ? מה הסיבוכיות ?
מחליפים אותו עם האחרון
מוציאים את האחרון החדש
מחלחלים את השורש עד שמוצא לו מקום חדש
log n

איך עושים "הוסף איבר" לערימה בינארית ? זמן ?
מוסיפים אותו למטה ו "מפעפעים" אותו למעלה
O (log n)

איך "מסיירים" במימוש בין הצמתים של ערימה בינארית ?
ע"פ אינדקסים. בנים הם 2n ו 2n+1

בניית ערימה make_heap

מהו האלגוריתם והסיבוכיות ליצירת ערימה
מקלט שרירותי ?
בניית מערך לא ממוין כעץ בינארי : O(n)
סידור העץ לערימה, heapify איבר איבר
שורה שורה מלמטה, ליתר דיוק מהאיבר ה
n/2 עד 1, ולכן אין צורך לגעת ברמה
התחתונה. O(n)

למה לכאורה נראה שהאלגוריתם ליצירת
ערימה הוא בסיבוכיות n log n ? מהי
ההוכחה לכך שהסיבוכיות היא אכן log n ?
כי המקרה הגרוע הוא שכל איבר עולה עד
השורש ועובר עד log n פעולות.
עושים סיגמא מ 1 עד אינסוף של i-1 חלקי
2^i , שסכומו קטן שווה מ 2n
או
חלוקת סיגמת הביטוי הכללי
(n / 2^logn) (log n)
לשורות ש "מתקטנות"
ולחסום כל שורה

האם בכדי לבנות ערימה ממערך צריך
לעשות heapify מלמטה או מלמעלה ? למה
?
למטה.
כי השגרה מחליקה ערכים למטה, אז
אחרת, אם יש איבר גדול למטה, הוא לא
יכול לעלות ?

האם ניתן לבנות ממערך בלתי ממוין ערימה
בזמן לינארי ? למה ?
כן
יש לחשב סיבוכיות מדויקת ולצמצם עם
נוסחאות טורים

איזה צומת נבדק ראשון ב make_heap ?
למה ?
n/2
אין צורך ביותר מזה. בצורה זו כבר נגיע לכל
העלים, גם אם הם מקס'.

אילו עלים בדיוק צריכים להבדק בבניית
ערימה ?
פנימיים. שאינם עלים.

עץ בינומי

מהו האלגוריתם האינטואיטיבי ליצירת עץ בינומי ?
הכפלת העץ
"הרמת" ההעתק הימני
קישור הראש הימני לראש השמאלי

מהו האלגוריתם הפורמלי לבניית עץ בינומי ?
B0 - צומת בודד
B1 – איחוד של שני עצי B0
Bk – איחוד של שני עצי Bk-1 כאשר מחברים את שורשו של אחד העצים לשורש השני

האם כל עץ בינומי מוגבל להגדרה הרקורסיבית הספציפית, על פיה העץ תחת השורש זהה לעץ תחת הבן השמאלי ?
כן

האם תכונת חביבות הבנים נשמרת בכל העץ הבינומי, כמו בערימה, או רק בשורש ?
בכל העץ

כמה צמתים מכיל עץ בינומי ? למה ?
2^k
ראה אלגוריתם אינטואיטיבי

האם כל העצים הבינומים מאותה רמה זהים ? למה ?
כן ?
מעצם הגדרת יצירתם ?

מה יפה בדרגות הבנים ?
כל בן הוא העץ המקורי לעומקו

מהו k של עץ בינומי Bk ?
גובה השורש ?

האם יש קשר תוכני בין שני עצים בינומים ?
לא בהכרח

מהו זמן איחוד שני עצים בינארים ? ובינומים ? למה ?
O(m+n) – מבוצע איבר איבר
O(log(m+n) - מבוצע עץ עץ ?

כמה איברים בעץ Bk ? מה עקרון ההוכחה ?
2^k
אינדוקציה

מהו גובהו של עץ Bk ? למה ? הוכחה ?
K

כי בכל הכפלה בעצם מוסיפים רמה אחת למעלה
אינדוקציה

האם ניתן להוסיף איבר אחד לעץ בינומי ? למה ?
על הפשט – לא, כי הערימה גדלה בחזקות של 2
רק לעץ בינומי ריק..

כמה ילדים לשורש של ערימת Bk ? למה ?
מה עקרון ההוכחה ?
אין ערימת Bk יש רק עץ Bk ?
K
כי בכל שלב גדל מספר הילדים של השורש ב 1
אינדוקציה

למי בעץ יש הכי הרבה ילדים ? למה ?
לשורש, כי רק לו נוספים ילדים, וכולם היו פעם שורש אז יש להם פחות..

מהי סיבוכיות איחוד שני עצים בינומים זהים ? למה ?
O(1)
כי צריך רק לקשר את השורשים ?

ערימה בינומית

מהי ערימה בינומית ?
מספר עצים בינומים מסודרים לפי הגודל השורשים מחוברים ברשימה מקושרת

מה ההבדל בין עץ בינומי לערימה בינומית ?
ערימה בינומית מורכבת ממספר עצים בינומים שונים, המסודרים לפי הגודל

מהו האלגוריתם האינטואיטיבי לבניית ערימה בינומית ?

איך מוסיפים איבר לערימה בינומית ?
עוברים עץ עץ לפי הגודל. העץ הראשון שלא קיים נוצר בעזרת קודמיו והאיבר החדש (1). זאת אומרת, אם עץ 1 לא קיים – מכניסים אליו. אם 2 לא קיים – יוצרים אותו בעזרת 1 והאיבר החדש.

מבחינת המימוש - הוספת קישור נוסף משורש הערימה הימנית, אל שורש הערימה השמאלית
בסוף אפשר להשוות כל שלב לייצוג הבינארי של העצים.

האם ייתכנו שתי ערימות בינומיות שונות מחזקה גבוהה זהה ? למה ?
כן. מעצים שונים ?
לא ? כי הייצוג הבינארי הוא אחד ?

מהו יער בינומי ?
אוסף של עצים בינומים, אשר עבור כל K לכל היותר עץ B_k אחד
אין יער. יש ערימה ?

האם ייתכן מספר זוגי של -- ? למה ?
לא

איך מקושרים העצים הבינומיים ?
ישנו שירשור קישורי בין השורשים השורשים מהווים רשימה מקושרת

איך תראה ערימה בינומית מ 23 צמתים ?
בוחרים אילו עצים לבנות ע"י הייצוג הבינארי של 23

האם הייצוג הבינארי למספר הצמתים הוא חח"ע ? למה ?
כן
כי לא ניתן לייצג מספר באופן בינארי בשתי דרכים

מהי סיבוכיות איחוד שתי ערימות בינאריות ? למה ?
$\log n$
כמו חיבור שני מספרים בינאריים

מהו יחס החביבות בין שורשי העצים הבינומים ?
אין בהכרח יחס כלשהו

מהי "הוכחת הטרמינטור" ?
המחשה לאלגוריתם הוצאת איבר.
הטרמינטור יורה מכוון b_0 והעץ הראשון שעומד מת ומתחלק לכל אלו ששכבו לפניו

מהי סיבוכיות בניית ערימה בינומית ? למה ?
$O(n)$
לכאורה $n \log n$ אבל כשסוכמים מגלים שמצטמצם.. כמו ערימה בינארית

כיצד ניתן לייעל מציאת חביב בערימה בינומית ל $O(1)$?
להוסיף פוינטר לחביב, ובהוצאה למצוא את החדש בסיבוכיות $O(\log n)$

האם ניתן להוציא מערימה איבר שאינו שורש ? למה ?
לא
זו ההגדרה

איך מאחדים שתי ערימות בינאריות ? כמה זמן זה לוקח ?
שלוש אפשרויות :
= איבר איבר, $O(m \log n)$
= בניית חדשה, $O(n + m)$
= איחוד עצים $O(\log (m+n))$

מהי סיבוכיות הוספת איבר לערימה בנומית ? למה ?
$O(\log n)$
מציאת מקום ריק $O(\lg n)$
סידור העצים מחדש $O(\log n)$

מהי "שיטת הפוטנציאל" להוספת איבר לערימה בינומית ?

ערימת חציון

מהו חציון ?
האיבר בעל האינדקס האמצעי בקבוצה
ממוינת

מהי "ערימת חציון" ?

מהו תרגיל ערימת החציון ?
נתונה רשימה מקושרת של מספרים
ממשיים, וידוע גם ערך החציון. יש לממש
את הפעולות הבסיסיות

מהו המימוש לאתחול ערימת החציון ?
הוספת איבר ? הוצא איבר ? אחזר חציון ?
הוצא חציון ?
שתי ערימות בינומיות הפוכות, Small
ממוינת max ו Large ממוינת min.
החציון תמיד בראש Large

מהי סיבוכיות כל פעולה ?
אתחול O(n)
הוספה O(log n)
הוצאה O (log n)
החזרת ערך חציון O(1)

מיונים

על אילו מיונים נסב הדיון ? מהם הסוגים ?
מהו הזמן הגרוע עבור כל אחד ?
מיון דלי bucket
מיון בסיס - Radix sort $O(n * k)$
מיון מניה
<u>מיוני השוואה n^2</u>
מיון בועות bubble
מיון ריצה
מיון הכנסה n^2 insertion
מיון בחירה selection
מיון מקסימום max
<u>מיוני השוואה n log n</u>
מיון מהיר quick
מיון ערימה heap
מיון מיזוג

להוסיף טבלת תכונות מדף עזר למבחן
יציבות, רציפות, במקום

מהו קירוב סטרלינג ?
קירוב מתמטי ל !n עבור מספרים גדולים

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

מהו מיון יציב ? למה זה חשוב ?
מיון שבו סדר איברים זהים נשמר בין הקלט לפלט
למיון איברים שיש להם מספר מאפיינים,
לפי מאפיין בודד כלשהו

אילו מיונים לא ניתן לממש כיציבים ? למה ?
Heap sort
Quick sort

מהו מספר הספרות בבסיס b של מספר x ?
Log x בבסיס b

מהי הסיבוכיות של שינוי בסיס למספר
?למה ?
O(1)

מהו סכום הטור
Log1 + log2 + log3 +… + log n
?
Log (1 * 2 * … * n)
Log(n!) =
Log n^n
n log n ?

Differential
נגזרת

מהו מיון במקום ?
מיון שסיבוכיות המקום שלו היא 1 ?

מהו מיון רציף ?
קשור לאיברים זהים ?

אילו מיונים אינם רציפים ? למה ?
מיון ערימה ועץ חיפוש מאוזן ?

מהו יחס סדר מלא ?
סדר לינארי
סדר חלקי המקיים גם את תכונות ההשוואה.
בסה"כ 4 התכונות :
רפלקסיביות
אנטי סימטריות
טרנזיטיביות
השוואתיות או טוטאליות : ניתן להשוות בין
כל שני איברים

למה פונקציות המיון מקבלות גם מקום
התחלה וסוף ? יש מערך – שימיינו ! לא ?
התחומים מאפשרים את הקריאות
הרקורסיביות

מה חשוב לזכור כשמחשבים זמן עבור 2
מערכים ?
להתחשב ביחסי הגדלים בין המערכים
לבחירת האלגוריתם המיטבי.

נתונים n איברים. הראה כיצד למצוא את
האיבר ה-K
(K-1 איברים גדולים ממנו) ב O של Klogk
פעולות. הנח שK קטן בסדר גודל מ-n.
?

לאילו מיונים עדיפות במקרה של התפלגות
פעמון גאוס ?
מיון דלי ?

אילו מיונים רצים בפחות מ nlogn ? מה
התנאים להפעלת כל אחד מהן ?
מיון בסיס radix
מיון דלי bucket
?

מה השיקולים לבחירת מיון השוואה ?

מיוני השוואה של n^2

מהם מיוני השוואה של n^2 ?
מיונים בהם בליבת הפעולה החוזרת
מבוצעת השוואה בין שני איברים :
מיון בועות
מיון ריצה
מיון הכנסה
מיון מקסימום

איך מוכיחים שמיוני השוואה הם לפחות
בסיבוכיות teta(nlogn) ?
עצי השוואה

מהם עצי השוואה ?
תרשים זרימה המתאר את ריצת אלגוריתם
מיון השוואה
התרשים נראה בסופו של דבר כמו עץ

מהו מספר העלים ומהו גובה עץ השוואה ?
זמן הריצה ?
n!
h = nlogn

מהו אלגוריתם מיון הכנסה ?
עוברים על המערך משמאל לימין.
עבור כל איבר מזיזים אותו שמאלה עד
שמגיע למקומו במערך הממוין הנוצר לאיטו
משמאל

מהו אלגוריתם מיון בחירה ?
= מצא את האיבר בעל הערך הנמוך ביותר
במערך
= החלף אותו עם האיבר הראשון במערך
= המשך באותה שיטה על שאר המערך
(ללא האיבר הראשון)
זו כנראה הדרך האינטואיטיבית ביותר למיין,
אך כאמור היא איננה יעילה במיוחד
במונחים של זמן ריצה.

מהי סיבוכיות מיון דלי במקרה הממוצע ?
האם זה שונה במקרה הגרוע ? למה ?
O(n) ?
כן
?

מיוני השוואה של n log n

מהם המיונים הרלוונטיים ?
מיון מהיר quick
מיון מיזוג merge
מיון ערימה Heapsort

מה זה heapify ? איך זוכרים ?
פונקציית הפיכת עץ בינארי לערימה בינארית.
משמשת עם כפונק' עזר למיון ערימה
המילה היא על משקל modify, "הפיכה לערימה.."

מהי שיטת "הפרד ומשול" ?
מתייחסת למיון מיזוג.
למעשה מדובר בשלושת השלבים הרקורסיביים :
הפרד לבעיות קטנות
פתור כל בעיה בנפרד
צרף חזרה לבעיה הגדולה

למה צריך merge בתוך שגרת מיון מיזוג ?
זוהי פונקציית האיחוד. הדרך הביתה חזרה..

מהי שגרת מיון מיזוג ?

```
mergesort(Array m)
{
if length(m) ≤ 1
return m
else {
middle = length(m)/2
for each x in m up to middle
add x to left
for each x in m after middle
add x to right
left = mergesort(left)
right = mergesort(right)
result = merge(left, right)
return result } }
```

מהי נוסחת הרקורסיה עבור מיון מיזוג ?
למה ?

$$T(n) = 2T(n/2) + O(n)$$

2 = תתי בעיות בתוך כל מופע בלולאה
n / 2 = מספר בעיות בתוך כל תת בעיה
O(1) = זמן ביצוע כל חלוקה
O(n) = זמן ביצוע הצירוף

מהו אלגוריתם מיון ערימה ?
התיאור הבא מתייחס ל"ערימת מקסימום", שבה כל הורה גדול בערכו משני בניו.

איך עובד מיון ערימה ?

במילה – בונים ערימה וחוזרים על extract max עבור כל האיברים. בפירוט :
= בונים ערימה ממערך של איברים למיון. O(n)
= מוציאים מהערימה את השורש (המהווה את הערך המקסימלי), ומכניסים אותו למיקום הפנוי הגדול ביותר במערך החדש. מתקבל מבנה של 'כמעט ערימה' הקטן בגודלו באיבר אחד.
= מוציאים את האיבר האחרון במבנה (שהיה אחד הערכים הקטנים בערימה) ומציבים אותו במקום השורש שהוצא.
= מבצעים פעולת Heapify שמטרתה לתקן את המבנה ולהפוך אותו שוב לערימה חוקית. מבצעים זאת באמצעות פעולה מחזורית מהשורש ומטה. O(log n)
= חוזרים שוב על תהליך הוצאת השורש עד אשר הערימה מגיעה לגודל של איבר בודד והמערך מכיל את כל איברי הערימה ממויינים. O(n log n)

האם מיון מהיר יכול לקחת O(n^2) ?
כן. במקרה הגרוע

מהו אלגוריתם מיון מהיר ?
אלגוריתם רקורסיבי הפועל בשיטת הפרד ומשול. צעדיו הם כדלקמן:
= בהינתן סדרת איברים, בחר איבר מהסדרה באקראי (נקרא: pivot, או "איבר ציר").
= סדר את כל האיברים כך שהאיברים הגדולים מאיבר הציר יופיעו אחרי האיברים הקטנים מאיבר הציר.
= באופן רקורסיבי, הפעל את האלגוריתם על סדרת האיברים הגדולים יותר ועל סדרת האיברים הקטנים יותר.
= תנאי העצירה של האלגוריתם הוא כאשר ישנו איבר אחד, ואז האלגוריתם מודיע כי הסדרה ממוינת.

במיון מיזוג, עבור מספר אי זוגי של איברים, איזה מערך יהיה גדול יותר ?
זה לא באמת משנה.
מקובל שהשמאלי

מיונים שאינם השוואה

אילו מיונים אינם מיוני השוואה ? למה ? מה סיבוכיות כל אחד ?
מיון דלי bucket O(n^2)
מיון בסיס radix O(n*k)
מיון מניה counting

איך עובד מיון מניה ?
שדה האיברים שיש למיין כולל ערכים שהם מספרים טבעיים בטווח 1->k. מערך המקור, אותו יש למיין, ייקרא A, ומערך המטרה, לתוכו תישמר תוצאת המיון, ייקרא B .
יוצרים מערך C בעל k תאים מאופסים.
עוברים על מערך A ומונים את מספר המופעים של כל איבר בו. כלומר, כאשר מגיעים לאיבר A[i]=j , אזי מבצעים את הפעולה C[j]+=1. לאחר שהסתיים המעבר על מערך המקור, התא C[i] מכיל את מספר ההופעות של איבר i במערך A. לאחר מכן, סוכמים את מספר המופעים - מוסיפים לכל C[i] את קודמו C[i]+=C[i-1] .
באופן זה, בתא C[i] שמור מספר האיברים אשר קטנים או שווים ל- i במערך A. בשלב הבא עוברים על מערך A מהסוף להתחלה, ועל כל איבר מבצעים:
B[C[A[i]]] = A[i]
C[A[i]] -= 1
המערך B יהיה ממוין, והסדר בין איברים שווים נשמר. זאת משום שבתא C[i] נשמר המופע האחרון המיועד של האיבר i במערך B, ומשום שהמעבר על מערך A היה מהסוף להתחלה.
מה שונה בין מיון מניה פה ל bucket sort ? למה ?
במיון מניה שנלמד במבנה ישנה לולאת סכומים
בכדי לייצב את המיון

איך עובד מיון בסיס ?
= מיון כל המספרים בקבוצה ל־10 קבוצות, על פי ספרת האחדות שלהם.
= מיון כל המספרים מהקבוצות שהתקבלו שוב, ל־10 קבוצות, על פי ספרת העשרות של כל מספר, תוך כדי שמירה על סדר המספרים בתוך כל קבוצה, לפי המיון של השלב הקודם.
= המשך מיון המספרים על פי הספרה החשובה יותר (מאות, ואז אלפים, ואז עשרות אלפים וכן הלאה), תוך שמירת סדר המספרים בכל קבוצה לפי המיון של השלב הקודם לשלב הנוכחי.

מדוע מבצעים radix sort מהעמודה הימנית ?
זה חוסך זכרון שיש להקצות

אם הולכים משמאל לימין אז נוצר בלאגן בגדלים. ה MSB חייב להיות ממוין אחרון.

מהי הדרך היעילה למיין n מספרים שלמים בתחום 0 עד n^2 - 1
? למה ? מהי הסיבוכיות ?
מיון בסיס על בסיס n
O(n)

איך ממיינים מיון בסיס על בסיס n ?
מעבירים את כל האיברים לבסיס n
מפעילים מיון בסיס. לכל מספר עד 2 ספרות

איך נראה מערך C בסוף מיון מניה ? ממוין ?

טבלאות גיבוב \ ערבול hash tables

מהי טבלת גיבוב ?
Hash table
טבלת ערבול

מה ההבדל בין טבלת דילוג לטבלת גיבוב לטבלת ערבול ?
אין הבדל בין טבלת גיבוב לערבול
אין טבלת דילוג. יש רשימת דילוג. ראה skip list.

מהי טבלת מיעון ישיר ?
טבלה ללא דילוגים. שבה ניתן להכניס לכל מפתח איבר אחד בלי התחכמויות..
אין התמודדות עם התנגשויות

מה מקור המילה גיבוב ?
מרמז על חיתוך וערבוב אקראיים

איך נקרא מצב שבו ערך רוצה להכנס לתא תפוס ?
התנגשות

מהי התנגשות ?
אם x1 שונה מ x2 אבל H(x1) = H(x2)

מהם הפתרונות להתנגשויות ?
= שרשור, chaining – כל תא ב T מצביע לרשימה מקושרת. Chain hashing
= הפנייה מחודשת לפי פונק' אחרת כלשהי

מהו סדר הגודל לפעולות על טבלה ברשימה מקושרת ?
האלמנט הארוך הוא תמיד המציאה
מקרה ממוצע : find (x) – O(1)
Find(x) : worst case teta(n)
אם הפיזור שווה ואז O(1)

מהי שיטת החילוק

מה ההבדל בין טבלת גיבוב פתוחה לסגורה ? איך זוכרים את זה ?
התמודדות עם התנגשות
טבלה סגורה משרשרת
טבלה פתוחה מדלגת

את מבינה למה קוראים לטבלה משרשרת סגורה ולמדלגת פתוחה ?
טוב. כדי לזכור החלטתי שפתוחה יכולה "לצאת" מתא המפתח המקורי שלה ומשרשרת לא אז היא סגורה
מה יהיה m גודל טבלה, ביחס למספר באיברים שעליה לאכסן ?

האם : מחלקה H היא מחלקה אוניברסלית של פונקצית גיבוב ? מה המשמעות ?
כן

מהי "הנחת הפיזור האחיד הפשוט" ?
simple uniform hash
h מפזרת את המפתחות באופן אחיד על פני m התאים

מתי עדיפה טבלה פתוחה ומתי סגורה ?
הטבלה הסגורה אינה טובה בהוצאת איברים

מהי בעית היחידות ? מה הקשר לטבלת ערבול ?
נתון מערך מספרים. צריך להגיד אם קיימים בו שניים זהים.
מיון יקח n log n
הכנסה לטבלת ערבול פתוחה והשוואה בכל התנגשות ייתן O(n)

מהו פקטור העומס ? מה יהיה שווה בטבלה פתוחה ?
מקדם העומס הוא מספר האיברים פר תא קטן שווה אחד

האם פונקציית גיבוב קיימת לטבלה סגורה או פתוחה ? למה ?
גם וגם
השינוי הוא בהתנגשות שמתרחשת אחרי פונק הגיבוב

מהי "הקבוצה האוניברסלית של המפתחות" ?
מרבית הפונק מניחות שמדובר בטבעיים ?

מהם "חיפוש כושל" ו "חיפוש מוצלח" ?

פונקציות גיבוב

מהן הדרישות מפונ' גיבוב ?
= דטרמיניסטיות : לכל ערך בתחום יתקבל
תמיד אותו ערך בטווח
= ניתן לחשב ב O(1) במקרה הממוצע
= מפזרת היטב
= קלה לחישוב

מהן 2 השאלות החשובות בגיבוב ?
כיצד לבחור פונ' גיבוב טובה
מה עושים במקרה של התנגשות

כיצד בוחרים את המחלק בשיטת השארית
?
טוב שיהיה ראשוני שאינו קרוב לחזקה של
2. חזקות קרובות של 2 גורמות לפיזור לא
אחיד כאשר המפתחות כתובים בבסיס
שהוא חזקה של 2, למשל מחרוזות תווים
נכתבות בבסיס 28 = 256 .

האם רק לטבלה פתוחה ישנן פונק גיבוב ?
לא. גם לשרשורים צריך להגיע איכשהו..

מהי פונק' ערבול ?
hash : h: U -> (0,…,m-1)
אשר בהינתן מפתח בתחום U מחשבת
אינדקס בטווח המתאים. האינדקס של
מפתח k הוא h(K).

Gcd
Great common devider
מחלק משותף גבוה

מהן פונק' הגיבוב הפופולריות ?
שארית, שיטת החילוק
שיטת הכפל

מהי שיטת ה xor ?
פונק ערבול

מהי שיטת הכפל ?
h(k) = [m(kA mod 1)]
מכפילים ב A שבין 0 ל 1, לוקחים את חלק
השבר ומכפילים ב m ואז שומרים את החלק
השלם

טבלת גיבוב פתוחה

מהם סוגי הפתרונות להתנגשויות בטבלה פתוחה ?
סריקה ליניארית linear probing– מעבר ליניארי על התאים עד למציאת התא הבא הפנוי
ערבול נשנה
ערבול כפול

מה הבעיה בסריקה לינארית ? מה הפתרון ?
לא ניתן להוציא איבר אם איבר עם מפתח זהה שהוזז לפניו הוצא כבר. לא נשארו "עקבותיו"
בזמן הוצאה להוציא את כל המוזזים ולהכניסם מחדש אחריה. או,
שיטת המציבה – תא שהוצא ממנו איבר מקבל ערך או דגל זמני "delete"

מהו מספר טוב לשיטת הכפל ? למה ?
ע"פ knuth : (sqrt5 – 1)/2 ~ 0.618

מתי בשיטת החילוק פונק ההאש לוקחת את הביטים הפחות משמעותיים ? מה זה אומר ?
m = 2^p או m = 10^p
זה פחות טוב כי מצמצם את האינפורמציה במקום לערבב את המספרים ?
רוב המספרים יגיעו לאותה שארית ולא יהיה פילוג טוב

מהי טבלת Open addressing ?
כשלא נעשה שימוש בשרשראות, אלא כל האיברים יוכנסו לטבלה.

איך דואגים לכך ש m לא יתחלק בפונ' הצעד ? מה המשמעות ?
בחירת m ראשוני ו r(x) חזקה של 2 למשל

מהו "ערבול נשנה" rehashing ?
נניח שברשותנו סדרה אינסופית של פונקציות ערבול: h0,h1,h2,…
ננסה לשמור את x במקום h0(x). אם תפוס, ננסה במקום h1(x). נמשיך עד שנצליח.

מהו ערבול כפול ?
נגיע לתוצאות דומות לערבול נשנה ע"י שתי פונקציות בלבד d, h.
כאשר: hi(x) = h(x) + i d(x)
הפונקציות d, h נבחרות באופן בלתי תלוי.

מהו היחס הרצוי בין d(x) לגודל הטבלה m ?
גודל הטבלה ו- d(x) צריכים להיות מספרים זרים כך ש $h_0(x),\ldots,h_{m-1}(x)$ תכסה את כל האינדקסים האפשריים בתחום {0,...,m-1}.
לפיכך נוח לבחור את m להיות מספר ראשוני.

מהו גיבוב קוקיה ?
שתי פונקציות ערבול
אם יש התנגשות גם בתא השני, מוציאים משם את הערך הקיים ובודקים עבורו את הפונקציה השנייה

מתי הומצא גיבוב קוקיה ?
2007 ?

ערבול אוניברסלי

מהו גיבוב אוניברסלי ?
פתרון גיבובי.
כמו שיטת הכפל או החילוק

איזו בעיה מנסה הערבול האוניברסלי לפתור ? מה הפתרון ?
לכל בחירה של פונקציית ערבול קיימת סדרה גרועה של מפתחות כך שתווצר רשימה באורך מקסימלי.
תכונה זו יכולה ליצור בעיה.
לדוגמא, יתכן מתכנת המשתמש באופן עקבי בשמות מסוימים למשתני התוכניות שהוא כותב ולצערו
פונקצית הערבול הבונה את ה- symbol table ממפה את כל השמות הנ"ל לאותו המקום בטבלת הערבול.
לפיכך **כל תוכנית מחשב** של משתמש זה אינה יעילה כפי שיכולה הייתה להיות !
<u>הפתרון</u>: לבחור באקראי, בזמן **יצירת** טבלת ערבול, פונקצית ערבול מתוך קבוצת פונקציות שהוגדרה מראש. נרצה שקבוצת הפונקציות תהיה כזו, שעבור כל סדרת מפתחות, בחירה אקראית של אחת הפונקציות תיצור פיזור טוב.

מהי universal hashing ?
עבור כל זוג מפתחות מהתחום הסיכוי להגיע לאותו מפתח הוא 1/m .
כאשר m הוא מספר התאים בטבלה.

איך יוצרים universal hashing ?
יש כמה שיטות :
ax = b mod n עבור n ראשוני
מספרים בני 64 ביט

מהי קובצה אוניברסלית ?
<u>הגדרה:</u> תהי H קבוצת פונקציות ערבול מתחום U לקבוצה {0,...,m-1}.
הקבוצה H נקראת **אוניברסלית** אם לכל זוג מפתחות שונים $x,y \in U$ מספר הפונקציות עבורן h(x) = h(y) הוא |H|/m.

האם עבור פונקציה אוניברסלית מתקיים – מספר ההתנגשויות הצפוי לכל מפתח קטן מגורם העומס.
? למה ? מה המשמעות ?
כן

עוד נושאים

מה זה העניין עם החלוקה לחמישיות ?
מיון ? מציאת חציון ?

מה הסבירות שתהיה שאלה במבחן של רישום פסאודו קוד ?
גבוהה מאוד ?

איך מסומן מצביע ל NULL ברשימה מקושרת ?
כמו הארקה

מה זה אלגוריתם ?
תהליך חישובי מוגדר היטב המקבל ערכי קלט ומפיק ערכי פלט

אינדנטציה
הזחות ?

פסאודו-קוד ?
קוד גולמי.
עקרון בצורה המוגדרת המופשטת ביותר

מהי שגרה בהקשר הקורס ?
קוד תוכנה

RAM
Random access machine

מהי נוסחת נסיגה ?
רקורסיה ?

מהי "פרדיגמת הפרד ומשול" ?

מה ההבדל בין טור לתור ?

מהי טבלת גיבוב ?

מהו תכנון דינמי ?

למה בספר אין גרפים ובהרצאות כן ?
יש גם בספר

מה משמעות המושג "החזר" ?
המצביע שהפונקציה מחזירה ?

מהו RMQ ומיהם האון ופישר ?
לא למדנו

מהו האתגר אצל רחל ?
גם פורמליות .. ?

מהן השגרות המותרות לשימוש עבור כל מבנה נתונים ?
build
insert
delete object
delete struct
find_max
find
join
split
נוספות :
עץ א"ש –
OS_rank
ערימות –
union
deunion

LCA ?

LCS ?

חציונים וערכי מיקום

איך מוצאים חציון ב O(n) ?
שיטת החמישיות
אלגוריתם SELECT

מהו הרעיון של אלגוריתם החמישיות ?

מדוע אלגוריתם החמישיות אינו בעצם
הפניה מעגלית ?

איך מוצאים אם איבר קיים במערך יותר מ
n/2 פעמים, בזמן O(n) ?
מוצאים חציון וסופרים מס' הופעותיו במערך

מהו חציון ממושקל ?
סכום איברים לפני ואחרי שווה

מהו אלגוריתם החמישיות למציאת חציון ?
+ חלוקה לחמישיות O(n)
+ מציאת איבר אמצעי בכל חמישיה O(n)
+ מציאה חציון y של קבוצת החציונים,
המספר שהתקבל קטן וגדול מ 3/10
מהאיברים בקבוצה השלמה
+ השווה את כל האיברים בקבוצה ל y.
+ מצא לאיזו קבוצה שייך y.
+ אם לא שייך לקבוצה אז r = y.
+ בצע שוב את האלגוריתם עבור הקבוצה
הקטנה יותר.

מדוע באלגוריתם החמישיות מחלקים דווקא
ל 5 ?
למספר זוגי אין אמצע
ל 3 נוצרת בעיה ?

קוד הופמן

מהו קוד הופמן ?
קוד הפמן הוא שיטה לקידוד סימנים, כגון תווי טקסט, המספקת דחיסת נתונים מרבית, כלומר מאחסנת את הסימנים במספר מזערי של סיביות. השיטה מתבססת על אורך משתנה לסימנים, כך שסימן נפוץ יוצג באמצעות מספר קטן של סיביות. זוהי גרסה כללית יותר של עקרון קידוד הנקרא "קידוד אנטרופיה".
עץ תחיליות
עץ בינארי שמסלול בו מתואר ע"י ביטים.
אחד עבור ימין, אפס לשמאל.

מהם השימושים של קוד הופמן ?
העברת טקסט בצורה "דחוסה".

כיצד צריכות להיות מסודרות התדירויות בעץ ?
שכיחות נמוכה במקום נמוך

מהו קוד תחיליות אופטימלי ?
קוד הופמן שבו העלים מסודרים לפי השכיחות, מנמוכה למטה לגבוהה למעלה ?

מהו עקרון ההוכחה לכך שהסידור היעיל ביותר הוא זה שבו העלים היא לא שכיחים הם הכי עמוקים ?
מציאת ביטוי פורמלי למחיר

מה צריך לדעת למבחן על קוד הופמן ?
כנראה שלא הרבה..

איך יודעים איפה מסתיימת אות בקידוד בינארי של מילה בקוד הפמן ?

מהו הרעיון האלגוריתם ?
נבנה את העץ הבינארי של הקוד "מלמטה למעלה". ראשית, נמצא את שתי האותיות שמספר המופעים שלהם **מינימלי**, וניצור צומת חדש, כך ששתי האותיות הללו יהיו בניו. כעת נתייחס לצומת החדש בתור אות חדשה, שמספר המופעים שלה הוא סכום מספרי המופעים של שתי האותיות שהן בניה. כך קיבלנו צמצום של הבעיה מבעיה ב-n אותיות לבעיה ב-$n-1$ אותיות.
נחזור על התהליך עד שנישאר עם צומת אחד בלבד - ראש העץ.

prefix free ?
אף מילה אינה רישה של מילה אחרת

מה הקשר בין הופמן לדחיסה ?
זה הלב שלו. הוא חוסך מקום בקידוד הקוד

מה קורה אם לא יודעים את סטטיסטיקת החזרות של האותיות ?

איך ייתכן קוד שאינו prefix free אבל עדיין אפשר לקדד ?
אם ההמשך של המילה המקודדת שונה אז אפשר להבין שצריך להפריד ולא להשתמש במילה הארוכה

איך דוחסים תמונה בהופמן ?
כאילו כל צבע הוא אות

רשימות דילוג skip list

מהי רשימת דילוג ?
מבנה נתונים המחזיק שתי רשימות מקושרות. אחת מחזיקה את כל הנתונים והשנייה רק דוגמת איברים מידי פעם, ולכן יותר מהר לרוץ עליה

באנגלית ?
Skip list

מהי תדירות תחנות האקספרס ביחס ללא ?
שורש n

מהי עלות החיפוש האופטימלית ?
2 sqrt n

איפה ממומשות ?
סאבווי ניו יורק

מהו הפיתוח המתבקש ?
הוספת רמה נוספת ועוד רשימה מקושרת

מה יהיה זמן החיפוש עבור שלוש רשימות ?
3 sqrt n

מהו אורכה הכללי של הרשימה הכי קצרה ?
(n sqrt) n

מהו מספרן המירבי של הרשימות ?
Log n

מהו זמן החיפוש הכללי ? מה ההוכחה ?
O(log n)

מהו האלמנט הרנדומלי בבניית מבנה הנתונים ?
בחירת תחנות אקספרס

האם SL שקול לעץ אדום שחור ?
בסופו של דבר מבחינת הסיבוכיות זה יוצא דומה ?

מהי סיבוכיות הכנסה ?
Log n

מהו האלמנט האקראי ?

תכנון דינאמי

מהו תכנון דינאמי ?

מהם העקרונות לתכנון דינאמי ?
1 + פתרון רקורסיבי פשוט
2 + סיבוכיות אקספוננציאלית
3 + בניית מבנה נתונים המכיל את כל הפתרונות האפשריים
+ מציאת פתרון = O(1)
+ הפתרון מוריד את הסיבוכיות לפולינומיאלית
4 + חשב את הטבלה בתור טבלה.
+ בניית טבלה O(n^2)

מהו סוג השאלות לתכנון דינאמי ?

מה הקשר בין תכנון דינאמי לרקורסיה ?
יחס הפוך – מגדול לקטן לעומת מקטן לגדול
תנאי ההתחלה שקול לתנאי העצירה

מהם שלבי פתרון הבעיה ?
1 + אפיון מבנה הפתרון – לנסות מטריצה חד מימדית. אם לא ניתן – מט' דו מימדית.
2 + הגדרה – מה יהיה בכל תא בכלל ובתא האחרון במטריצה : A[I,j]. דגש על האבחנה האם הנוסחה מחייבת שימוש בתא האחרון או לא
3 + הגדרת נוסחת התכנון הדינאמי
4 + הגדרת תנאי ההתחלה
5 + הגדרה – באיזה תא במטריצה ימצא הפתרון
6 + מציאת סיבוכיות עבור מילוי כל תא בפרט, ועבור כל הטבלה בכלל
7 + הוכחת הפתרון ע"י אינדוקציה
8 + אלגוריתם

איך מזהים שאלות בתכנון דינאמי ?
לא צריך להמציא מבנה נתונים

מה החסרון של תכנון דו מימדי ?
סיבוכיות של n^2

מהי נוסחת התכנון הדינאמי ?
בדרך מקס' או מינ' כלשהו
מתאר את השגרה להתקדמות הרקורסיבית

מהו אלגוריתם חמדני ?
אלגוריתם רקורסיבי ?
מנסה לתת פתרון פשוט למצב "לוקלי" לכל המצבים
פתרון "פרגמטי" לפי כלל אצבע שרואה רק את המצב הנוכחי אבל לא זוכר הסטוריה מעבר

מהו הייתרון של עבודה עם תכנון דינאמי ?

סיבוכיות נמוכה מרקורסיה

מהי סיבוכיות הוספת איבר ומילוי הטבלה בבעיית חיתוך הקרש ?
O(k)
O(k^3)

באיזו מטריצה נשתמש עבור שאלת תת הרצף ? מה הגדרתה ?
חד מימדית
כל תא – הרצף המרבי עד אליו שכולל אותו.
נוסחת התכנון הדינמי – האם להוסיף את הערך בתא או לא, או פשוט לשים את הערך עצמו לבד

האם לבעיית הרצף אפשר להגדיר מערך שבו כל אינדקס משמעותו אורך רצף, ובתוכו הרצף המירבי ?
לא, משום שאין פה אלמנט רקורסיבי ?

מהו backtrace ? איך משיגים אותו ?
להגיע לדרך שבה הגענו לתשובה בתכנון דינאמי
שומרים פוינטרים

מהו אלגוריתם חמדן ?
[אלגוריתם](#) המתבסס על [היוריסטיקה](#) בה בוחרים את האפשרות הטובה ביותר הניראית לעין, מבלי לקחת בחשבון את השלכות הבחירה לטווח רחוק.

מהי הדוגמא הקלאסית להוכחת חמדנותו של האלגוריתם ?
הסוכן הנוסע – תכנון נתיב עי החוקיות "סע לנקודה הקרובה ביותר שלא היית בה"..

על איזו שאלה שלא רשומה בבדח תכנון דינמי צריך תמיד לענות ?
האבחנה האם הנוסחה מחייבת שימוש בתא האחרון או לא

איך מתמודדים עם שוויון בין אפשרויות בנוסחאות הרקורסיביות ?

איך יודעים האם נוסחת הרקורסיה צריכה לרוץ כל פעם מתחילת המערך או רק לבדוק מספר קבוע של איברים אחרונים ?
אם המידע על האיברים הראשונים "מוכל" באחרונים אז כנראה שלא צריך לחזור על הכל ?

איך מתמודדים עם מצב ששני פתרונות "מתחרים" על המיטביות, וצריך "לזכור" את שניהם, כמו שתי סדרות עולות עבור שאלת ה LIS ? האם זה קיים גם ל LCS ?
?
לא ?

האם ניתן לכתוב גם אלגוריתם חמדני בצורת פתרון דינאמי ?
כן ?

מהן השאלות המכוונות אל עבר נוסחת הרקורסיה ?
"מהן האפשרויות החדשות שלי לצעד החדש"?

מהי "שיטת התזכור ?"

מה היו ההתקעויות הגדולות שלי בפתרון שאלות בתכנון דינאמי ?
כשחשבתי שאפשר עם מערך חד מימדי אבל התשובה היתה במטריצה

מהו פתרון איטרטיבי ?

מהם סוגי התשובות בתכנון דינמי ?
פתרון איטרטיבי
שיטת התזכור ?

מהן הבעיות הקלאסיות של תכנון דינאמי ?
תת סדרה רצופה גדולה ביותר
בעיית הקטעים 1 – מירב קטעים זרים לא חופפים
בעיית הקטעים 2 -

מהו הפתרון החמדני לבעיית הקטעים 1 ?
בוחרים כל פעם את הקטע המסתיים ראשון
מוחקים את כל הנחתכים איתו
חוזרים על התהליך

מבוא לאלגוריתמים

כיצד מוכיחים נכונות של אלגוריתם ?

מהן הדרישות מאלגוריתם ?
נכונות
סיבוכיות יעילה : זמן, מקום,

מהו פרדוקס האצת האלגוריתם הקבוע ?
אפשר להאיץ כל אלגוריתם נתון פי קבוע
אז למה לא ניתן להאיץ כבר בסדר גודל ?

איך תמיד אפשר להאיץ אלגוריתם ?
להגדיל את הזכרון שבו נעשה שימוש בתוכנית

O notaition ?
סדר גודל
O

כמה מקום בזכרון תופס מספר כלשהו ? איך זה בא לידי בחישובי הסיבוכיות ?
log x
תלוי בגודלו
צריך לזכור שגודל הקלט אינו בהכרח שווה למספר הקלטים

כמה זמן לוקחת פעולה חשבונית פשוטה ?
עבור מספרים קטנים פעולות חשבוניות לוקחות זמן קבוע
הסכמה עבור הקורס הנוכחי

מהו אלגוריתם יעיל ? למה ?
זמן ריצה פולינומי באורך הקלט
ז"א
עד n^k
סיבות פרקטיות – ניתן לבצע כאלו אלגוריתמים עבור קלט הגיוני

מהי קופסא שחורה ?
שימוש בפתרון של תת בעיה כשלב באלגוריתם הכללי

איך יודעים איזו פעולה פשוטה יותר ?
אם ניתן לבצע אליה רדוקציה

מהי רדוקציה ?
מעין הפשטה, ירידה במורכבות פעולה

מהו משפט המאסטר ?

אילו חידות שואלים בראיונות עבודה ?
חידת הזהב והעודף
חידת 1000 החולים ?
סדרת פיבונאצ'י

מהו פתרון נאיבי ?
הפתרון הטריוויאלי

מה ההבדל בין o קטן לגדול O ?

אילו גדולים יש ואילו קטנים ?
3 גדולים – O, omega,

מהי פעולה אטומית ?

כמה ביטים נדרשים לייצוג אונרי של מיליון ?
מיליון

מהי מטריצת שכנויות ?
טבלה שבה העמודות והשורות הן קודקודים, ויש 0 או 1 אם יש או אין קשת בין כל שני קודקודים

מה זאת אומרת – הסיבוכיות צריכה להנתן כתלות באורך הקלט ?

מה זה טריסאט ?

איך מאכסנים גרף ברשימה מקושרת ?

מה רושמים באלכסון של מטריצת שכנויות ?
תלוי בשאלה שמנסים לפתור

מה זה פריקסות ?

מה זה over kill ?
ביצוע פעולות חישוב מיותרות או בחירת אלגוריתם נאיבי אבל לא מיטבי

מהן שתי הגישות הכלליות למציאת אלגוריתם על סמך אלגוריתמים ידועים ?
להתייחס לאלגוריתמים הידועים כקופסא שחורה
להכנס לאלגוריתם ידוע ולשנות אותו

מהו אלגוריתם חמדן ?
זהו אלגוריתם שתמיד לוקח את הפתרון הטוב ביותר המיידי על מנת למצוא פתרון.

טרנספורם פורייה

fft ?
fast fourier transform

שימושים ?
עיבוד תמונה, סינון אותות
הכפלת מספרים גדולים ב nlogn
הכפלת פולינומים

ייצוג ע"י ערכים לעומת ייצוג ע"י מקדמים

מה זה בעצם ?
בחירת הערכים עבורם תבוצע המכפלה
למציאת וקטור הערכים עפ וקטור המקדמים

כל מספר הוא פולינום ?
כן. פולינום המקדמים של הבסיס שלו

מערכת סינגולרית ?
מטריצה הפיכה

ערכים p מקדמים alfa ?
כן

מהי מטריצת ונדרמונדה ?
מטריצת המומנטים
דטרמיננט שלה – מכפלת ההפרשים, שונה מאפס
מטריצה שמכילה את כל שורות alfa מחזקה אפס עד חזקה d

איך מוצאים את וקטור הערכים ?
מטריצה ההפוכה לונדרמונדה, כפול וקטור המקדמים

מה מיוחד בערכים הנוחים ?
הם מרוכבים

כמה ערכים צריך למצוא מקדמים לפולינום מדרגה d ? למה ?
d+1

איך קוראים ל x בפולינום ?
alfa1 -> n

מהו d ?
החזקה הגבוהה

שורש יחידה ?
w^n = 1

מהו שורש היחידה עבור 4 <- n=1 ?
1 : 1
1-,1 : 2
1cis0,120,240 : 3
1, i, -1, -i : 4

מהם המספרים המיוחדים של פורייה ?
עבור איזו דרגת פולינום ?
n שורשי היחידה השונים, במקום האלפות
n-1

מהו הסידור הגנרי ?

מהי ההגדרה ?
TF של פולינום p מדרגה קטנה מ n הם ערכי p ב n שורשי היחידה השונים מסדר n

מהו שורש יחידה פרימיטיבי משורש n ?
אם הוא לא שורש יחידה גם מסדר קטן יותר כלשהו

איזה אלגוריתם נלמד ? מהי שיטתו ?
?
חלוקת פולינום לשני פולינומים
ציור עץ
עלי FT
חזרה למעלה בעץ לקבלת פולינום חדש ?

תחקירי גיחות

בעבודה על שני מערכים אפשר למיין רק את הקטן, וכך להקטין את גורם הלוגריתם שבסיבוכיות

המטרה היא להשתמש כמה שיותר במושגים שנלמדו

הוספת ערך קבוע לכל הקשתות עלול לשנות את המסלול הזול ביותר, כי תלוי באורכו.