שאלה 1

פתח אלג' יעיל לבעיה הבאה. קלט: גרף לא מכוון G=(V,E)
צומת v∈V ומספר k. מצא (אם קיים) עץ פורש של G
שבו ל-v דרגה בדיוק k.

פתרון ינון :
אם k גדול מדרגתו הנוכחית של v לא ניתן לעמוד בתנאי השאלה. אם לא -
נפרוש את G ע"י DFS מ v.
נבדוק מהי דרגת v בעץ הפורש. נקרא לה 'k.
אם k < 'k אז ניתן לפרוש את G כשדרגת v היא k.
אלגוריתם – נוסיף שרירותית קשתות מ v עד שדרגתו תהיה k,
ונמחק את הקשתות הנכנסות לאותם צמתים.

ניתוח של סרגיי :
השאלה שייכת למשפחת שאלות: נתון גרף וקבוצת קודקודים A בגרף (גודל הקבוצה קבוע).
רוצים לבנות עץ פורש (מינימאלי אם יש משקלים) כך שדרגת הצמתים בקבוצה תהיה בדיוק (לכל היותר, לכל הפחות) K.
שימו לב, שכאשר גודל הקבוצה אינו קבוע, הבעיה היא NP קשה ולא ניתן לפתור אותה באופן אופטימאלי בזמן פולינומי.
שאלה דומה נשאלה בתרגיל בית – שם רצינו שהצמתים הנבחרים יהיה עלים בעץ פורש – כלומר דרשנו דרגה == 1.
שאלה אחרת – שאלת מבחן שרבים פתרו בזמן ההכנה – רוצים עץ פורש עם בדיוק K קשתות אדומות. קל לעשות רדוקציה משאלה 1 לשאלה זו – פשוט צובעים את כל הקשתות שיוצאות (נכנסות) מצומת V באדום ומפעילים את האלגוריתם למציאת עץ עם בדיוק K אדומים.
הנה פסאודו קוד של אלגו' שאילן נתן בשעת קבלה:

```
R=red e(u,v) in G
B=black e(u,v) in G
G'=G-R
FIND ALL Ri
Ri= מחלקת קשירות ב G'
Find ALL Ti
Ti= עץ פורש ב Ri


W(e) for all e IN R = 1;
W(e) for all e IN B = 2;

Ti=u in G'''
FIND  MST IN G'''+R (improve MST max k times).

Coplexity is O (KN)
```

הנה עוד שאלה בסגנון עם רמז:
Let *G* be a connected, undirected graph with real-valued edge weights, let *r* be a distinguished vertex of *G*, and let *k* be an integer. A *k-tree* of *G* is a spanning tree of *G* in which vertex *r* has degree *k*. Devise an efficient algorithm to compute a minimum-total-weight k-tree of *G* if one exists. Prove the correctness of your algorithm, and analyze its running time.

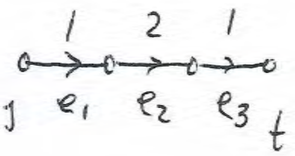
Hint: Prove that (if a minimum k-tree exists) there is a minimum k-tree all of whose edges are either incident to *r* or in a minimum spanning tree of $G-\{r\}$. Thus prove that by means of a single minimum spanning tree computation, the k-tree problem can be reduced to a k-tree problem on a subgraph *G′* of $O(n)$ edges, consisting of the union of a tree and the set of edges incident to *r*. Prove that a minimum k-tree can be obtained from a minimum $k+1$ (or $k-1$) tree by doing a single edge swap (adding one edge and deleting one other edge). Using this result, show how to find a minimum k-tree in *G′* fast. (A minimally acceptable solution is $O(n \log n)$ time; there is a way to do it in $O(n)$ time.)

שאלה 2

תהי $G=(V,E)$ רשת זרימה עם קיבולות שלמות $c:E\rightarrow \mathbb{N}^+$ מקור s ובור t.

לכל קשת e נגדיר $\Delta(e)$ השינוי (ההקטנה) הדרושה ב- $c(e)$ כך ש- e תהיה בחתך מינימום s-t ברשת (אם אין אופן כזה, נגדיר $\Delta(e)=\infty$).

דוגמא: ברשת



$\Delta(e_1)=\Delta(e_3)=0$

$\Delta(e_2)=1$

פתרון ינון :
נמצא זרימה מרבית בגרף ע"י EK.
נעדכן עבור כל קשתות הגרף ערך התחלתי : d(e) = c(e)-f(e).
נצייר את הגרף השיורי.
- נבחר קשת. נחבר את שני קצותיה, אחד למקור והשני לבור, עם קשתות חדשות שקיבוליהן אינסוף.
- נבדוק מחדש את הזרימה המקסימלית. זמן : EK.
- אם הזרימה לא גדלה, אז הקשת היא חלק מחתך ונסמן סופית את d(e)
- אם הזרימה גדלה, אז הקשת אינה חלק מחתך.
- נקטין את קיבולת הקשת והזרימה באחד, ונבדוק שוב את הזרימה המקסימלית.
- אם הזרימה המקסימלית כעת שווה למקורית, אז הקשת היא חלק מחתך.
- נחזור על התהליך עד שנגיע לקיבולת אפס. מספר החזרות המרבי שווה לזרימה המרבית F.
- אם הגענו לקיבולת אפס אז הקשת לעולם לא תהיה חלק מחתך ולכן d(e) שווה אינסוף.
- ניתן שם לבדיקת הקיבולת לחתך : "בדיקת חיתוך לקשת"
- נבצע את בדיקת החיתוך לקשת עבור כל קשתות הגרף. נכפיל את הסיבוכיות ב E.
סה"כ הזמן :
EK * E * F
רעיון נוסף :
עבור כל קשת בגרף נבדוק מהו הערך שעבורו היא קריטית להקטנה.
אם היא אינה קריטית להקטנה, נקטין את הקיבולת והזרימה עליה ב 1,

ניתוח של סרגיי :

מובן לכולם שכדי לפתור את השאלה צריך לבחון את הרשת השיורית (RESIDUAL) אחרי שמוצאים זרימה מקסימאלית (ניתן אפילו להניח שהזרימה נתונה – זה לא עיקר האלגוריתם). האלגוריתם צריך בעצם לחלק את הקשתות ל3 קבוצות (חלקן אולי ריקות). קשתות שכבר בחתך מינימום כלשהוא, קשתות שע"י שינוי הקיבולת שלהן הופכות לחברות בחתך מינימום כלשהוא, וקשתות שאין להן סיכוי להיות באף חתך מינימום.

אני מזכיר שעשינו דברים דומים:
ראינו איך ניתן לבדוק שקשת נמצאת בחתך מינ' – ניתן היה להשתמש ברעיון כקופסא שחורה.
דיברנו על התסריט של שינוי קיבולת של קשת והשינויים שזה גורם לזרימה בכל הרשת. גם זה היה יכול להיות כלי באלגוריתם שלכם.

בשאלה זו הסיבוכיות אינה חשובה כל כך. תשובות שרצו זמן לינארי ולא כיסו את כל המקרים כמעט ולא קבלו נקוד. לעומת זאת תשובות נכונות עם הסבר מפורט קבלו את רוב הנקודות גם כשהן רצות זמן ריבועי.

למשל, היו אנשים שרק בדקו אם הקשת רוויה או לא. פתרון זה אמנם יעיל, אבל לא תמיד נכון. הוא אף פעם לא יחזיר ערך אינסוף לקשת ויכול לתת ערכי Δ שגואים.

שאלה 3

עבור עץ $T=(V,E)$, קבוצה $S \subseteq V$ נקראת __שולטת__ אם לכל $v \in V$, או $v \in S$ או קיים $u \in S$ ו $(u,v) \in E$ (או שניהם).
כלומר, S שולטת אם לכל $v \notin S$ יש שכן ב-S.
כתוב אלג' אשר בהינתן עץ מוצא קבוצה שולטת בעלת גודל קטן ביותר

פתרון ינון :
נבחר צומת שרירותית בגרף שאינה עלה ונשתמש בה לשורש העץ
נמצא את כל העלים בעץ.
נבחר ל S את כל הצמתים שהם אבות של עלים בעץ.
עבור כל עלה, נמשיך לטפס בעץ בדילוגים באורך של 3 קשתות (דילוג על שני צמתים), ואת הצומת השלישי בספירה נוסיף ל S.
נמשיך בטיפוס עד שנגיע לשורש, או עד שנפגוש צומת מ S ואז נמשיך את טיפוס ממנו.

ניתוח של סרגיי :
שאלה קלאסית בתכנון דינאמי על עצים. הקבוצה השלטת הנאיבית לכל עץ הינה העץ עצמו. אנו רוצים למצוא קבוצה מינימאלית – כלומר קבוצה כזאת, שאם מסירים ממנה איבר, אזי נוצר צומת ללא שכן בקבוצה.
נאיבי – נסמן כל קשת ב 0 או 1 – אם הוא בקבוצה או לא. כל בחירת קבוצה הינו מספר בינארי – חלקם מייצגים קבוצות לא חוקיות. נעבור על כל האפשרויות ונמצא את האופציה החוקית הכי קטנה. זמן ריצה – אקספוננציאלי.
רקורסיה – נניח שכל הצמתים כבר סומנו באופן אופטימאלי פרט לצומת V.
 אם כל השכנים של V בקבוצה, אזי סיימנו – לא צריך להוסיף את V ויש לנו קבוצה אופטימאלית. אם אף שכן של V לא בקבוצה אז נוסיף את V ועדיין נקבל משהו אופטימאלי.

מה עושים במידה ורק חלק מהשכנים של V בקבוצה? איך דואגים לשמר את האופטימאליות?
מדוע אם שומרים מידע בצמתים האלגוריתם רץ בזמן פולינומי?

שאלה 4

כתוב אלגוריתם אשר בהינתן $G=(V,E)$, גרף לא מכוון, $e \in E$ ו- $x,y \in V$ מחליט האם יש ב-G מסלול פשוט מ-x ל-y המכיל את e.

רמז: מנגר.

ניתוח סרגיי :
בשאלה אתם נדרשים להחליט אם קיים מסלול פשוט שעובר בקשת נתונה. ניתן גם היה לבקש למצוא את כל המסלול – זה לא היה משנה כלל.
את השאלה ניתן לפתור בשתי דרכים:

א. יש למצוא את רכיבי הקשירות (צמתי הפרדה) בעזרת DFS. מתקבל גרף "קקטוס". כעת צריך לבחון את כל המקרים הבאים:

- הקשת ושני הצמתים באותו רכיב קשירות [3.0.0]
- הקשת ברכיב קשירות אחד והצמתים (שניהם) ברכיב קשירות אחר [2.0.1]
- הקשת ברכיב קשירות אחד והצמתים (שניהם) בשני רכיבי קשירות שונים [1.1.1]
- קשת וצומת אחד ברכיב באותו רכיב קשירות וצומת נוסף ברכיב אחר [1.0.2]

לגבי כל אחד מהמקרים צריך לתת נימוק מדוע קיים (או לא קיים) מסלול פשוט.

ב. ניתן לבצע רדוקציה לבעיית זרימה כפי שעשינו בכיתה. הקושי הוא להציג רדוקציה שבה:
קיום זרימה חוקית ברשת שבנינו ⇔ קיום מסלול פשוט כנדרש.
בפרוש ניתן לייצר רשתות זרימה לא נכונות ואז קיום זרימה לא יבטיח קיום מסלול פשוט!

מי שעשה ארבעה ריצות BFS (או בכלליות O(1) BFS-ים) פותר רק חלק מהמקרים, אבל לא את כולם – לרוב די קל למצוא דוגמה נגדית שתפיל את האלגוריתם הזה כי הקושי הטכני שבו נובע מחיתוך של מסלולים.