

מבני נתונים

תרגיל

100

1.

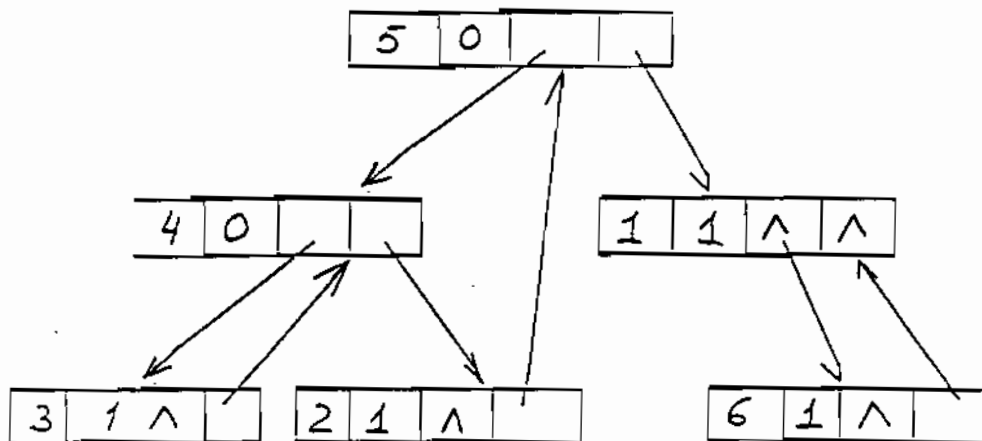
עץ כחווט (Threaded binary tree) הוא עץ בינארי שבו מצביע Right של קדקוד שאיזו לו בן ימני מראה על איבר הבא בסדר Inorder. בכל קדקוד קיים שדה Thread המגדיר:

- Thread = 0 אם לקדקוד יש בן ימני
- Thread = 1 אם לקדקוד אין בן ימני

כלומר אם Thread = 0 אז מצביע Right מראה על בן הימני ואחרת הוא מראה על איבר הבא בקדר Inorder.

Val Thread Left Right

דוגמא:



לאיבר אחרון בסדר Inorder מצביע Right שווה ל- NULL ( בדוגמה זה קדקוד עם ערך ...

כתוב פונקציה ב C++ המקבלת כפרמטר מצביע על שורש של עץ כחווט ומדפיסה את איברי העץ בסדר Inorder. הפונקציה צריכה להשתמש בחוטים. אין להשתמש ברקורסיה או מחסנית או הצביע אב. זמן ריצה של אלגוריתם הוא  $O(n)$  ( כאשר n מספר קדקודים בעץ )

2. נתונה רשימה מקושרת חד-כוונית באורך n שהיא ממוינת לפי מפתח k. הצע במנה נתונים יישתמש ב-  $O(m)$  מקום נוסף פרט לרשימה (  $m \leq n$  ) המאפשר גישה לאיבר עם מפתח k ברשימה מקושרת בזמן  $O(n/m + \log m)$ . הערה: הנח שמבנה כבר מאותחל ( זמן הבנייה שלו לא נכנס לזמן החיפוש ) ותוך מפעולות החיפוש לא מתבצעות פעולות אחרות על רשימה מקושרת.

3. רוצים להחזיק קטעים כגורים יבסי על הישר. כל קטע מיוצג ע"י קצותיו [a,b] שהם מספרים מסוג float. על המבנה הנתונים מוגדרות שתי פעולות:

insert(a,b) - מכניסה את הקטע [a,b] למבנה.

left(y) - מחזירה נקודה המינימלית x של קטע ש- y נמצא בו. אם y לא נמצא באף קטע מחזירה NULL.

האר מבנה נתונים למימוש הפעולות הנייל כך שכל פעולה תהיה ב-  $O(\log n)$  כאשר n הוא מספר קטעים במבנה. ( תאר גם כן את הפעולות )

הגשה: בזוגות עד ל- 12.12.96



```

inorder ( node * T;
{
while( T!= NULL)
{
while(T->Left != NULL )
    T = T-> Left;

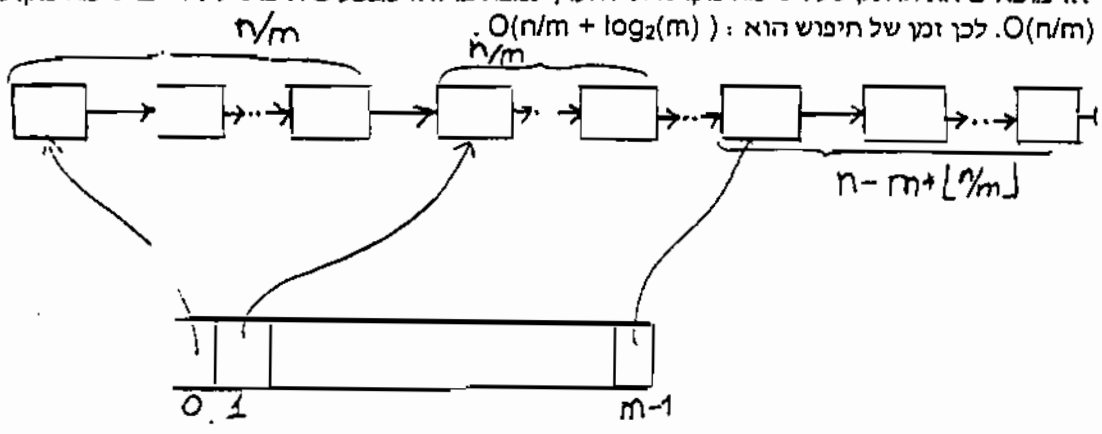
cout << T->Val;

while( T->Thread != 0 )
    { T = T-> Right; cout << T->Val ; }

T = T-> Right ;
}
}
    
```

~~AA as Thread~~

2. מחלקים רשימה מקושרת ל-  $m$  חלקים כך שכל אחד מכיל  $\lfloor n/m \rfloor$  איברים (חוץ מאחרון - הוא מכיל  $\lceil n/m \rceil$  איברים). מגדירים מערך בגודל  $m$  שמכיל מצביעים על תחילת כל חלק של רשימה מקושרת. כדי לבצע חיפוש קודם מתפשים ע"י חיפוש בינארי במערך - זמן החיפוש הוא  $O(\log_2(m))$ . אם לא מוצאים את הערך בין אברי המערך אז מוצאים את החלק של רשימה מקושרת שהערך נמצא בו ואז מבצעים חיפוש לינאר ברשימה מקושרת שזה לוקח  $O(n/m)$ . לכן זמן של חיפוש הוא:  $O(n/m + \log_2(m))$ .



3. מבנה הנתונים הוא עץ 2-3 כך שבעלים נשמור זוגות של מספרים וערך הראשון בזוג (הערך המינימלי) הוא המפתח (ז.א. הוא יופיע בתור מפתח בקודקדים הפנימיים).

insert(x,y)

- עושים חיפוש כמו בעץ 2-3 :

- אם  $k_1 > x, y$  אז ממשיכים חיפוש בתת עץ הראשון.
- אם  $x < k_1 < y$  אז קטע  $[x, y]$  לא יהיה זר לקטעים אחרים במבנה - מפסיקים חיפוש.
- אם  $k_1 < x, y$  (כאשר יש שני בנים) או  $k < x, y < k_2$  אז ממשיכים בתת עץ השני.
- אם  $x < k_2 < y$  אז קטע  $[x, y]$  לא יהיה זר לקטעים אחרים במבנה - מפסיקים חיפוש.
- אם  $k_2 < x, y$  ממשיכים בתת עץ השלישי.
- אם מגיעים לעלה והוא נוכיל קטע  $[a, b]$  שזר ל-  $[x, y]$  אז מייצרים קדקוד חדש עם  $[x, y]$  ומחזירים אותו לאבא של  $[a, b]$  - מכן כמו באלגוריתם עבור עץ 2-3.
- אם מגיעים לעלה שמכיל קטע  $[a, b]$  שאינו זר לקטע  $[x, y]$  אז לא עושים כלום.

Left(y)

מבצעים חיפוש של  $e$  כמו בעץ 2-3. כאשר מגיעים לעלה בודקים האם  $e$  נמצא תוך קטע שהיא בעלה. אם כן מחזירים גבול שמאלי של הקטע. אחרת מחזירים NULL.

