קורס 20407

סמסטר 2011א

מועד א׳ (87)

מבנה הבחינה: בבחינה חמש שאלות.

עליכם לענות על **ארבע** מתוך חמש השאלות.

לכל השאלות משקל שווה.

הנחיות: כל תשובה צריכה להתחיל בעמוד **חדש**.

אין לכתוב בצבע אדום.

אין לכתוב בעיפרון.

אפשר להשתמש בכל עובדה או תוצאה הנמצאת בספר הלימוד או במדריך הלמידה, ללא הוכחה או הסבר. חובה להוכיח או להסביר כל טענה אחרת.
אין צורך לכתוב פסֵידוקוד, אלא אם נדרש במפורש.

שאלה 1

נתונות שתי רשימות של מספרים ממשיים, S בת m איברים ו- T בת n איברים; בנוסף, נתון מספר ממשי z.

כתבו אלגוריתם הקובע האם קיימים $x \in S$, $y \in T$, כך שמתקיים $x + y = z$. זמן הריצה הנדרש של האלגוריתם הינו $\Theta((m+n)\cdot \lg(\min(m,n)))$.

פתרון:

נעתיק את שתי הרשימות למערכים (באותם שמות S ו- T).

נניח כי $m < n$. ממיינים את המערך S (מיון מיזוג או מיון ערמה); זמן ריצה $\Theta(m\cdot \lg m)$. לכל איבר $y \in T$ נבצע חיפוש בינרי במערך S אחר הערך $x = z - y$; זמן ריצה $\Theta(n\cdot \lg m)$. סה"כ $\Theta((m+n)\cdot \lg m)$.

אם $m > n$, דומה.

שאלה 2

א' (10 נקודות) נתון מערך של מספרים $A[1..n]$. ידוע שקיים אינדקס m , $1 < m < n$, כך שמתקיימים התנאים $A[1] > \ldots > A[m-1] > A[m] < A[m+1] < \ldots < A[n]$ (כלומר, התת-מערך $A[1..m]$ ממוין בסדר יורד והתת-מערך $A[m..n]$ ממוין בסדר עולה).

כתבו שגרת פסֵידוקוד המבצעת חיפוש אחר האינדקס m והמחזירה אותו; זמן הריצה הנדרש הינו $O(\lg n)$.

ב' (15 נקודות) נתון מערך של מספרים $A[1..n]$. ידוע שקיימים אינדקסים p, q , $1 < p < q < n$, כך שמתקיימים התנאים $A[1] > \ldots > A[p-1] > A[p] = \ldots = A[q] < A[q+1] < \ldots < A[n]$ (כלומר, התת-מערך $A[1..p]$ ממוין בסדר יורד והתת-מערך $A[q..n]$ ממוין בסדר עולה).

כתבו שגרת פסֵידוקוד המבצעת חיפוש אחר האינדקסים p, q והמחזירה אותם; זמן הריצה הנדרש הינו $O(\lg n)$.

פתרון:

א׳ השגרה בפסֵידוקוד :

```
FIND-MINIMUM(A)
left ← 1
right ← length[A]
while left < right
   do m ← (left + right) / 2
      if m = left
         then return m
      else if A[m] < A[m+1]
         then right ← m
         else left ← m
```

שאלה 3

א׳ (12 נקודות) נתון מערך $A[1..n]$ של מספרים שלמים המקיימים את התנאים

$n^2 \le A[i] \le n^3 + n^2 - 1$ לכל $i = 1,...,n$.

כתבו שגרה למיון המערך בזמן לינארי.

ב׳ (13 נקודות) נתון מערך $Q[1..n]$ של מספרים ממשיים ; ידוע שלא קיימים במערך יותר מ-

$\lg^2 n$ ערכים שונים זה מזה.

הראו שניתן להכניס כל איברים המערך לעץ אדום-שחור בזמן $O(n \cdot \lg\lg n)$.

פתרון:

א׳ בונים את המערך $B[1..n]$: $B[i] = A[i] - n^2$. אז מתקיים $0 \le B[i] \le n^3 - 1$ לכל $i = 1,...,n$.
ניתן לייצג כל איבר של B כמספר בן שלוש ספרות בבסיס n . משתמשים במיון בסיס מעל מיון-מניה.

ב׳ בונים את העץ האדום-שחור בהתחשב בכפילויות של המפתחות. מתקבל עץ בגודל $O(\lg^2 n)$ שגובהו $O(\lg(\lg^2 n)) = O(\lg\lg n)$.

שאלה 4

הציעו מבנה נתונים S התומך בפעולות הבאות בזמנים הנדרשים (n מציין את מספר האיברים במבנה) :

BUILD(S) : בניית המבנה S מסדרה של n מפתחות ; זמן הריצה : $O(n\cdot\lg n)$;

INSERT(S,k) : הכנסת המפתח k לתוך המבנה S ; זמן הריצה : $O(\lg n)$;

DEL-MEDIAN(S) : מחיקת החציון של המבנה S ; זמן הריצה : $O(\lg n)$;

MIN-OLDEST(S,t) : החזרת המפתח המינימלי בין t המפתחות הותיקים ביותר במבנה ; זמן הריצה : $O(\lg n)$.

הערה: המבנה S יכול להיות מורכב מכמה מבני נתונים פשוטים יותר.

פתרון: **כפי שראינו במפגש, הפתרון שגוי**

מבנה הנתונים S מורכב מערמת מינימום $H_{\min}$, ערמת מכסימום $H_{\max}$, עץ אדום-שחור מורחב T (עץ ערכי מיקום) ומחסנית P. ערמת המינימום $H_{\min}$ מכילה את $\lceil n/2\rceil$ המפתחות הקטנים היותר של המבנה, ערמת המכסימום $H_{\max}$ מכילה את $\lfloor n/2\rfloor$ המפתחות הגדולים יותר של המבנה. העץ T והמחסנית P מכילים כל אחד כל האיברים. המפתחות של T הם זמני ההכנסה. כל איבר בעץ T קשור לאיבר המקביל במחסנית P ולאיבר המקביל באחת הערמות באמצעות מצביעים דו-כיווניים. כל איבר z במחסנית מכיל מצביע $\min[z]$ אל האיבר בעל המפתח המינימלי הקודם לו (או לעצמו).

BUILD(S) : מציאת החציון בעזרת האלגוריתם SELECT ; חלוקת המערך בעזרת השגרה PARTITION ; בניית ערמת המינימום מהמפתחות הקטנים ובניית ערמת המכסימום מהמפתחות הגדולים ; בניית המחסנית באופן סדרתי, בהוספת השדה $\min[z]$ (אם המפתח הנכנס הוא הקטן ביותר, אז המצביע מופנה לעצמו ; אחרת, הוא מופנה אל המינימום עבור האיבר הקודם ; עד עכשיו, זמן ריצה לינארי. בונים את העץ T בזמן $O(n\cdot\lg n)$.

INSERT(S,k) : הכנסה לעץ T ולמחסנית P, כרגיל, עם הוספת שדה המינימום ; האיבר שנכנס לעץ T מקבל מפתח גדול ב-1 מהקודם. הכנסה לאחת הערמות לפי המקרה (מפתח קטן מהחציון או גדול ממנו) ; העברת איבר בין שתי הערמות, לפי הצורך. זמן ריצה $O(\lg n)$.

DEL-MEDIAN(S) : מחיקת שורש ערמת המינימום ; העברת איבר בין שתי הערמות, לפי הצורך ; מחיקת האיברים המקביליים מהעץ ומהמחסנית. זמן ריצה $O(\lg n)$.

MIN-OLDEST(S,t) : מציאת ערך המיקום ה-t בעזרת OS-SELECT(T,t) ; מעבר אל המחסנית ובחירת $\min[z]$.

שאלה 5

הציעו מבנה נתונים S התומך בפעולות הבאות בזמנים הנדרשים (n מציין את מספר המפתחות השונים ב- S ; N מציין את המספר הכולל של מפתחות ב- S):

$\text{BUILD}(S)$: בניית המבנה S מסדרה של N מפתחות לא בהכרח שונים; זמן הריצה:
$O(N \cdot \lg n)$;

$\text{INSERT}(S,k)$: הכנסת המפתח k (מופע חדש או חוזר שלו) למבנה S ; זמן הריצה: $O(\lg n)$;

$\text{MAX-FREQ}(S)$: החזרת המפתח בעל השכיחות (כפילות) המכסימלית במבנה S; זמן הריצה:
$O(1)$;

$\text{SMALLER-KEYS}(S,k)$: החזרת מספר המפתחות במבנה S, הקטנים מהערך הנתון k (לכל מפתח סופרים את הכפילות שלו); זמן הריצה: $O(\lg n)$.

הערה: המבנה S יכול להיות מורכב מכמה מבני נתונים פשוטים יותר.

פתרון:

מבנה הנתונים S מורכב מעץ אדום-שחור מורחב T (עץ ערכי מיקום) ומעץ אדום-שחור רגיל F.
כל צומת z ב- T מכיל את שדה $freq[z]$ (השכיחות של המפתח) ואת השדה $size[z]$ (מספר המפתחות בתת-עץ המושרש ב- z, כולל כפילויות). כל צומת בעץ T מכיל מצביע לצומת מקביל בעץ F, וגם בכיוון ההפוך. המפתח ב- F הוא השכיחות של המפתח ב- T.

$\text{INSERT}(S,k)$: מחפשים את המפתח k בעץ T ;

אם הוא נמצא (בצומת z), מוסיפים 1 לשדה $freq[z]$; מוסיפים 1 לשדה $size[z]$ בכל האבות הקדמונים של z ; מוסיפים 1 למפתח המקביל ב- F, מוחקים את הצומת ומכניסים אותו מחדש;
אם הוא לא נמצא, יוצרים צומת חדש ב- T בעל מפתח k וכפילות 1 ויוצרים צומת חדש ב- F בעל מפתח 1; אם יש צורך לבצע סיבובים ב- T, משתמשים בנוסחאות

$$size[y] \leftarrow size[x]$$
$$size[x] \leftarrow size[left[x]] + size[right[x]] + freq[x]$$

מוסיפים 1 לשדה $size[z]$ בכל האבות הקדמונים של z ; זמן ריצה $O(\lg n)$.

$\text{BUILD}(S)$: מבצעים N פעולות הכנסת מפתח; זמן הריצה הכולל $O(N \cdot \lg n)$.

$\text{MAX-FREQ}(S)$: עוברים מהצומת בעל המפתח המכסימלי ב- F אל המפתח המקביל ב- T ;
מחזירים את מפתח הצומת; זמן ריצה $O(1)$.

$\text{SMALLER-KEYS}(S,k)$: מבצעים את פעולת החיפוש בעץ T אחר המפתח k ; בכל צומת z לאורך מסלול החיפוש אוגרים את המספר $size[left[z]]$ (אם $left[z]$ קיים; מחזירים את סכום שדות אלה; זמן הריצה $O(\lg n)$.

ב ה צ ל ח ה !