אפשר להשתמש בכל עובדה או תוצאה הנמצאות 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

נתונה נוסחת הנסיגה

$$T(n) = a \cdot T\left(\sqrt{n}\right) + \lg^2 n$$

כאשר $a > 0$ הינו פרמטר ממשי.

פתרו את נוסחת הנסיגה עבור הערכים האפשריים השונים של a.

פתרון:

נשתמש בשיטת החלפת המשתנים : $m = \lg n,\ \ n = 2^n$. מתקבלת נוסחת הנסיגה

$$T(2^m) = a \cdot T\left(2^{m/2}\right) + m^2$$

אחרי הסימון $S(m) = T\left(2^m\right)$, מתקבלת הנוסחה

$$S(m) = a \cdot S\left(m/2\right) + m^2$$

נשתמש בשיטת האב. ההפרדה למקרים השונים מתבצעת לפי הערך $\log_b a = \lg a$.

1) אם $a > 4$, אזי $\lg a > 2$, אנחנו במקרה 1, ומתקבל הפתרון $S(m) = \Theta\left(m^{\lg a}\right)$;

2) אם $a = 4$, אזי $\lg a = 2$, אנחנו במקרה 2, ומתקבל הפתרון $S(m) = \Theta\left(m^2 \cdot \lg m\right)$;

3) אם $a < 4$, אזי $\lg a < 2$, אנחנו במקרה 3, ומתקבל הפתרון $S(m) = \Theta\left(m^2\right)$.

עבור נוסחת הנסיגה המקורית, מתקבלים הפתרונות

1) אם $a > 4$, $T(n) = \Theta\left(\lg^{\lg a} n\right) = \Theta\left(a^{\lg\lg n}\right)$;

2) אם $a = 4$, $T(n) = \Theta\left(\lg^2 n \cdot \lg\lg n\right)$;

3) אם $a < 4$, $T(n) = \Theta\left(\lg^2 n\right)$.

שאלה 2

הציעו מבנה נתונים S התומך בפעולות הבאות בזמנים הנדרשים:

$\text{BUILD}(L,S)$: בניית המבנה S מתוך רשימה נתונה L בת n מפתחות; זמן הריצה: $O(n)$;

$\text{INSERT}(S,k)$: הכנסת המפתח החדש k למבנה S; זמן הריצה: $O(\lg n)$;

$\text{MEDIAN}(S)$: החזרת חציון המפתחות של S; זמן הריצה: $O(1)$;

$\text{DEL-MEDIAN}(S)$: מחיקת חציון המפתחות של S; זמן הריצה: $O(\lg n)$;

$\text{DECREASE-KEY}(S,p,d)$: הקטנת המפתח שאליו מצביע p בערך d; זמן הריצה: $O(\lg n)$.

הערה: המבנה S יכול להיות מורכב מכמה מבנים בסיסיים.

פתרון:

מבנה הנתונים S יהיה מורכב מערמת מכסימום H_1 וערמת מינימום H_2.

$\text{BUILD}(L,S)$: מעבירים את הרשימה L למערך; מפעילים את האלגוריתם SELECT למציאת החציון, אחר-כך מבצעים חלוקה סביב החציון. מ-$\lceil n/2 \rceil$ האיברים הקטנים יותר בונים את ערמת המכסימום H_1 ומ-$\lfloor n/2 \rfloor$ האיברים הגדולים יותר בונים את ערמת המינימום H_2 (משתמשים בשתי הגרסאות של BUILD-HEAP); זמן הריצה: $O(n)$.

$\text{INSERT}(S,k)$: אם המפתח k קטן מהחציון (או שווה לו), מכניסים לערמה H_1; אחרת, מכניסים אותו לערמה H_2. אחר-כך, אם מספר האיברים ב-H_1 גדול מ-$\lceil n/2 \rceil$, מוחקים את השורש של H_1 ומעבירים אותו ל-H_2; אחרת, אם מספר האיברים ב-H_2 גדול מ-$\lfloor n/2 \rfloor$, מוחקים את השורש של H_2 ומעבירים אותו ל-H_1; זמן הריצה: $O(\lg n)$.

$\text{MEDIAN}(S)$: החזרת מפתח השורש של H_1; זמן הריצה: $O(1)$.

$\text{DEL-MEDIAN}(S)$: מחיקת השורש של H_1; אחר-כך, אם מספר האיברים ב-H_2 גדול מ-$\lfloor n/2 \rfloor$, מוחקים את השורש של H_2 ומעבירים אותו ל-H_1; זמן הריצה: $O(\lg n)$.

$\text{DECREASE-KEY}(S,p,d)$: אם p מצביע אל איבר של H_1, מקטינים את המפתח בערך d ומפעילים את השגרה MAX-HEAPIFY; אחרת (p מצביע אל איבר של H_2), מקטינים את המפתח בערך d ובודקים אם הוא עדיין גדול או שווה לשורש של H_1. אם כן, מפעילים את השגרה המקבילה ל- HEAP-INCREASE-KEY עבור ערמות מינימום; אחרת, מחליפים את האיבר עם השורש של H_1, מתקנים את H_1 באמצעות הפעלת השגרה MAX-HEAPIFY ומתקנים את H_2 באמצעות הפעלת השגרה MIN-HEAPIFY. זמן הריצה: $O(\lg n)$.

שאלה 3

א' (5 נקודות) נתון המערך $[3,0,2,4,5,8,7,6,9]$ כפי שהוא נראה **אחרי** ביצוע שגרת החלוקה PARTITION.

אילו מהאיברים שלו היו יכולים לשמש כאיבר הציר בשגרת החלוקה? נמקו את תשובתכם.

ב' (20 נקודות) שגרת החלוקה PARTITION מופעלת על המערך $A[1..n]$ ויוצרת את המערך $B[1..n]$.

נתון מערך הפלט $B[1..n]$. כתבו אלגוריתם, יעיל ככל שאפשר, למציאת כל האיברים שהיו יכולים לשמש כאיבר הציר בשגרת החלוקה. נתחו את זמן הריצה שלו.

פתרון:

א' אחרי ביצוע החלוקה, כל איברי המערך הקטנים מאיבר הציר חייבים להימצא לפניו, וכל איברי המערך הגדולים ממנו חייבים להימצא אחריו. האיברים $<4;5;9>$ מקיימים את הנדרש.

ב' סורקים את המערך B משמאל לימין; כל איבר $B[i]$ שהוא מכסימום בתת-מערך $B[1..i]$ מועתק למערך עזר L. סורקים את המערך B מימין לשמאל; כל איבר $B[i]$ שהוא מינימום בתת-מערך $B[i..n]$ מועתק למערך עזר R. מחזירים את כל האיברים הנמצאים גם ב- L וגם ב- R. מכיוון שהמערך L ממוין בסדר עולה והמערך R ממוין בסדר יורד, אפשר למצוא את האיברים האלה בזמן $O(n)$. זמן הריצה הכולל: $O(n)$.

שאלה 4

הציעו מבנה נתונים S התומך בפעולות הבאות בזמנים הנדרשים:

$\text{SEARCH}(S,k)$: חיפוש אחרי המפתח k במבנה S;

$\text{INSERT}(S,k)$: הכנסת המפתח k למבנה S;

$\text{DELETE}(S,z)$: מחיקת האיבר שאליו מצביע z מהמבנה S;

$\text{MEDIAN}(k_1,k_2)$: החזרת החציון של תת-קבוצת המפתחות $\{k : k_1 \le k \le k_2\}$ של S.

כל אחת מהפעולות צריכה להתבצע בזמן $O(\lg n)$.

פתרון:

נשתמש בעץ אדום-שחור מורחב (עץ ערכי מיקום) T.

$\text{SEARCH}(S,k)$: חיפוש רגיל בעץ אדום-שחור ; זמן הריצה : $O(\lg n)$.

$\text{INSERT}(S,k)$: הכנסה רגילה בעץ ערכי מיקום ; זמן הריצה : $O(\lg n)$.

$\text{DELETE}(S,z)$: מחיקה רגילה בעץ ערכי מיקום ; זמן הריצה : $O(\lg n)$.

$\text{MEDIAN}(k_1,k_2)$: בעזרת השגרה OS-RANK מוצאים את המיקום r_1 של k_1 ואת המיקום r_2 של k_2 ; בעזרת השגרה OS-SELECT מוצאים את הערך שדירוגו $\lfloor (r_1+r_2)/2 \rfloor$; זמן הריצה : $O(\lg n)$.

הערה : הפתרון מסתמך על ההנחה שהמפתחות k_1 ו- k_2 נמצאים במבנה S.

שאלה 5

נתון עץ אדום-שחור T ; נסמן ב- ln את גודל התת-עץ השמאלי של השורש וב- rn את גודל התת-עץ הימני של השורש.

האם היחס $ln < 128 \cdot rn$ מתקיים תמיד? הוכיחו או הביאו דוגמה נגדית.

הערה: גודל תת-עץ (בעץ אדום-שחור) הינו מספר הצמתים הפנימיים שלו.

פתרון:

נבנה את העץ האדום-שחור הבא : התת-עץ הימני הוא עץ שלם בגובה h ; כל הצמתים שלו שחורים. התת-עץ השמאלי הוא עץ שלם בגובה $2h$; הצמתים שלו צבועים לסירוגין, כל הצמתים ברמות הזוגיות שחורים, כל הצמתים ברמות האי-זוגיות אדומים. העץ המתקבל הינו עץ אדום-שחור חוקי.

גודל התת-עץ הימני הוא $rn = 2^h - 1$; גודל התת-עץ השמאלי הוא $ln = 2^{2h} - 1$.

אם מתקיים התנאי $ln < 128 \cdot rn$, אז $2^{2h} - 1 < 128 \cdot (2^h - 1) < 2^{h+7} - 1$; כלומר, $h < 7$.

מסקנה : התנאי הנתון לא מתקיים תמיד.