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

שאלה 1

פתרו את נוסחת הנסיגה שלהלן:

$$\begin{cases} T(1)=\Theta(1) \\ T(n)=(9/4)\cdot T\left(\sqrt[3]{n^2}\right)+\sqrt[4]{\lg^3 n}\cdot(\lg\lg n)^3\cdot\left(\sqrt[4]{\lg^5 n}+(\lg\lg n)^5\right) \end{cases}$$

שאלה 2

נתונה רשימת קלט בעלת n איברים. קוראים את האיברים ברשימה בזה אחר זה.

(15 נק׳) א. עומד לרשותנו זיכרון מחשב שגודלו $O(k)$, כאשר $k<n$. הסבירו כיצד ניתן למצוא בזמן $O(n+k\lg k)$ את k האיברים הגדולים ברשימה (בסדר ממוין).

(10 נק׳) ב. נניח עתה כי לרשותנו זיכרון בגודל $O(n)$ לקליטת האיברים. הסבירו כיצד ניתן למצוא את k האיברים הגדולים ביותר בזמן $O(n)$.

שאלה 3

הציעו מבנה נתונים S המבצע את הפעולות שלהלן בזמנים הנדרשים (n מסמן את המספר הכולל של איברים במבנה ו- m מסמן את מספר האיברים השונים זה מזה):

$\text{BUILD}(L,S)$: בניית המבנה S מרשימה **ממוינת** L בת n איברים; זמן הריצה: $\Theta(n)$;

$\text{INSERT}(S,k)$: הכנסת מפתח k לתוך המבנה S; זמן הריצה: $\Theta(\lg m)$;

$\text{DELETE-LAST}(S,k)$: מחיקת העותק שנכנס אחרון של המפתח k מהמבנה S (אם הוא קיים); זמן ריצה: $\Theta(\lg m)$;

$\text{MAX-FREQ}(S)$: החזרת השכיחות המכסימלית של איבר במבנה S; זמן ריצה: $\Theta(1)$.

הערה: מבנה הנתונים S יכול להיות מורכב מכמה מבנים יסודיים.

שאלה 4

צומת בעץ בינרי נקרא מאוזן אם גובה התת-עץ השמאלי שווה לגובה התת-עץ הימני.

בהינתן עץ בינרי T בן n צמתים, נניח שכל הצמתים בעץ מאוזנים פרט לשורש: ההפרש בין גובה הבן השמאלי שלו וגובה הבן הימני שלו הינו 2.

(10 נקי) א. מהו המספר n של צמתים כפונקציה של הגובה h של העץ T?

(15 נקי) ב. הראו כיצד ניתן לאזן את השורש בעזרת סיבוב אחד או שני סיבובים. איך משתנה גובה העץ? (אין חובה לשמור על איזון הצמתים האחרים.)

שאלה 5

הציעו מבנה נתונים S שבאמצעותו ניתן לבצע את הפעולות הבאות בזמנים הנדרשים (n מציין את מספר האיברים של S):

$\text{SEARCH}(S,k)$: חיפוש אחר המפתח k במבנה S; זמן הריצה: $O(\lg n)$;

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

$\text{DELETE}(S,p)$: מחיקת האיבר שאליו מצביע p מהמבנה S; זמן הריצה: $O(\lg n)$;

$\text{DECREASE}(S,k,d)$: הורדת הערך $d>0$ מכל המפתחות של S הקטנים מ-k; זמן הריצה: $O(\lg n)$;

$\text{INCREASE}(S,k,d)$: הוספת הערך $d>0$ לכל המפתחות של S הגדולים מ-k; זמן הריצה: $O(\lg n)$.