<u>סמסטר ב' 2023/24</u>

<u>מבני נתונים ואלגוריתמים</u>

<u>תרגיל בית מספר 2</u>

<u>שאלה 1</u>

נתחו את זמן הריצה של QuickSort על הקלטים הבאים. כלומר, עבור על אחד מהקלטים, הראו g(n) כך שזמן הריצה של QS הוא $\Theta(g(n))$.

א. מערך שכולו אחדות: [1,1,1,1……,1,1].

ב. מערך שבמקומות 1 עד n/2 הוא כולו אפסים, ובמקומות n/2+1 עד n הוא כולו אחדות: [0,0…,0,0,1,1,…,1,1].

ג. מערך שבמקומות 1 עד $n^{1/3}$ מכיל מספרים שונים ממוינים מקטן לגדול, ובמקומות $n^{1/3}+1$ עד n מכיל את אותו מספר, השווה לזה שב $A[n^{1/3}]$.

<u>שאלה 2</u>

רשמו עבור כל אחת מהטענות הבאות אם היא נכונה או לא נכונה, ונמקו את תשובתכם. אם תשובתכם חיובית אז עליכם לתאר אלגוריתם שפותר את הבעיה (אין צורך לכתוב פסאודו-קוד וניתן להסתפק בהסבר מילולי) ולהסביר את זמן הריצה שלו. אם תשובתכם שלילית אז עליכם להסביר מדוע.

א. נתון מערך A שבו n מספרים שלמים מתוכם $\sqrt{n}$ מספרים שגודלם לא ידוע וכל שאר המספרים (כלומר $n-\sqrt{n}$ מספרים) הם שלמים בתחום 1,…,n. אזי ניתן למיין את המערך בזמן $O(n)$.

ב. נתון מערך בגודל n של שלמים שונים (אין חסם על גודלם). המערך נקרא מערך פרבולי אם מתקיים:

- לכל $1 < i \le \left\lceil \frac{n}{2} \right\rceil$ מתקיים $A[i] \ge A[i-1]$
- לכל $\left\lceil \frac{n}{2} \right\rceil + 1 \le i < n$ מתקיים $A[i] \ge A[i+1]$

לדוגמה המערך [3,4,12,9,8,6] הוא מערך פרבולי.

<u>הטענות</u>:

(1) קיים אלגוריתם מבוסס השוואות שהופך מערך למערך פרבולי (שמכיל את אותם מספרים) בזמן $O(n)$.

(2) קיים אלגוריתם מבוסס השוואות שהופך מערך למערך פרבולי (שמכיל את אותם מספרים) בזמן $O(n^{1.01})$.

שאלה 3

יהי $A=[a_1,\dots,a_n]$ מערך לא ממוין של מספרים **שונים זה מזה.** נגדיר את החציון של A להיות האיבר שנמצא במקום ה $i=\lfloor(n+1)/2\rfloor$ במערך הממוין שמכיל את האיברים המקוריים. לדוגמה, החציון של המערך [8,4,1,10,2] הוא 4 משום שהמערך הממוין המתאים הינו [1,2,4,8,10] והמספר 4 נמצא באינדקס $3=\lfloor(5+1)/2\rfloor$. באופן דומה, החציון של המערך [8,5,1,3] הוא 3.

א. השלימו את הפסאודו-קוד הבא עבור הפרוצדורה MedianRecursive שבהינתן מערך A כקלט וגודלו n תחזיר את החציון במערך. הפרוצדורה עושה שימוש בפרוצדורה Partition שנלמדה בכיתה עבור אלגוריתם QuickSort. שימו לב שכיוון שהמספרים במערך שונים זה מזה, Partition(A,p,r) סדרת מחדש את איברי תת המערך A[p,…,r] ומחזירה אינדקס q כך ש $p\le q<r$, וכל המספרים בתת המערך A[p,…,q] **קטנים** מהמספרים בתת המערך A[q+1,…,r]. הפרוצדורה גם עושה שימוש בפרוצדורה max(A,i,j) שבהינתן מערך A ושני אינדקסים $i\le j$ מחזירה את האיבר המקסימלי בתת המערך A[i,…,j]. זמן הריצה של הפרוצדורה max(A,i,j) הוא $\Theta(s)$ כאשר $s=j-i+1$.

```
Median(A, n)  {
    k := ⌊(n+1)/2⌋
    return MedianRecursive (A, 1, n, k)
}

MedianRecursive(A, p, r, k)  {
    q := Partition(A, p, r)
    left_size := q-p+1
    if (left_size = k)
        return max(A, ___________________________________________)
    else if (left_size > k)
        return MedianRecursive(A, _________________________________)
    else
        return MedianRecursive(A, _________________________________)
}
```

ב. מה זמן הריצה של הפרוצדורה Median (במקרה הגרוע)? כלומר, מהי g(n) כך שזמן הריצה של הפרוצדורה הוא $\Theta(g(n))$? יש להראות שזמן הריצה הוא גם $O(g(n))$ וגם $\Omega(g(n))$.

ג. בסעיף זה בלבד נניח שהפרוצדורה MedianRecursive קוראת לפרוצדורה ThirdPartition(A,p,r) (במקום לפרוצדורה Partition) שפועלת באופן דומה; הפרוצדורה ThirdPartition מקבלת כקלט מערך A ושני אינדקסים $p<r$ ומסדרת את איברי תת המערך A[p,…,r] כך ש $\mathbf{q=p+\lfloor(r-p)/3\rfloor}$ וכל המספרים בתת המערך A[p,…,q] קטנים מהמספרים בתת המערך A[q+1,…,r]. נניח שזמן הריצה של ThirdPartition הוא $\Theta(s)$ כאשר $s=r-p$.
נתחו את זמן הריצה של הפרוצדורה MedianRecursive לאחר השינוי כפונקציה של n. כלומר, מהי g(n) כך שזמן הריצה של הפרוצדורה הוא $\Theta(g(n))$?

ד. נתון מערך A שבו n מספרים, והניחו לצורך פשטות ש n הוא כפולה של 3. ידוע ששני שליש מהאיברים במערך (כלומר, 2n/3 איברים) הם מספרים שלמים בתחום {1,…,100n} ושאר האיברים הם מספרים שלמים שקטנים מאפס (אין חסם על גודלם).
תארו אלגוריתם למציאת החציון של המערך A בזמן O(n) במקרה הגרוע, ונתחו את זמן הריצה של האלגוריתם שתיארתם. ניתן לתאר את האלגוריתם במילים, אך התיאור צריך להיות תמציתי ומדויק.
<u>שימו לב</u>: אין להשתמש בפרוצדורה ThirdPartition.

שאלה 4

אחד הסטודנטים בקורס הציע לשנות את אלגוריתם CountingSort אשר ראינו בכיתה כך שאפשר יהיה לראות באופן מידי שזמן הריצה שלו הוא $\Theta(n+k)$ (ואולי צריך לחשוב קצת יותר על הנכונות שלו).
השלימו את הקוד המצורף והסבירו את פעולת האלגוריתם. אין צורך לתת הוכחת נכונות פורמאלית, אך יש לתת הסבר בהיר במובן שאילו אתם הייתם קוראים אותו הייתם מבינים את פעולת האלגוריתם. למען הפשטות אתם יכולים להסביר את פעולת האלגוריתם תחת ההנחה שכל $i\in\{1,\ldots,k\}$ מופיע לפחות פעם אחת ב A.

```
Counting-Sort(A,n ,k)
for i  = 1 to k
   C[i] ← 0
for j = 1 to n
    C[A[j]] ← C[A[j]] + 1
i ← 1
for j  = 1 to (n+k-1)  {
    if (C[i] > 0) {
        A[ _______ ] ← i
        C[i] ← C[i] -1
    }
    else    i ← i+1
}
```

שאלה 5

כתבו פסאודו-קוד למימוש טיפוס מבנה הנתונים המופשט תור כפי שנלמד בכיתה ובתרגול, כאשר הוא ממומש באמצעות מערך "ציקלי" ("מעגלי") כפי שנתאר להלן. במימוש זה מערך בגודל MAXSIZE ושני אינדקסים Q.front ו Q.rear עבור מיקום שני הקצוות של התור. כאשר מכניסים איבר לתור, מקודם האינדקס Q.rear במקום אחד ושם מונח הערך המוכנס לתור. כאשר מגיע האינדקס Q.rear למקום האחרון במערך (כלומר כאשר ערכו הוא MAXSIZE), הצעד הבא מחזיר אותו לתחילת המערך.
כאשר מוציאים איברים מהמערך מוחזר האיבר שנמצא במקום Q.front והאינדקס Q.front מקודם ב-1.
כך, כאשר מוציאים איברים מהמערך הם "מפנים מקום" לאיברים חדשים. בדומה ל Q.rear, גם כאשר Q.frontמגיע לסוף המערך (כלומר כאשר ערכו הוא MAXSIZE) בהוצאת האיבר הבאה הוא יקודם לתחילת המערך. דאגו לבצע בדיקות כנגד overflow (כלומר, כאשר יש כבר MAXSIZE איברים בתור ומנסים להכניס עוד איבר) ו underflow (כאשר אין איברים בתור ומנסים להוציא איבר). ניתן להיעזר במשתנה נוסף Q.num_of_elements.

עליכם לממש את הפעולות הבאות:

- EMPTY(Q) : מחזיר TRUE אם התור ריק ו-FALSE אחרת.
- ENQUEUE (Q,x) : הכנסת x לסוף התור.
- DEQUEUE (Q) : הוצאת האיבר מתחילת התור והחזרת ערכו.
- FRONT (Q) : החזרת ערכו של האיבר בתחילת התור.