מבוא והוכחת נכונות של אלגוריתמים

נתון מערך A של מספרים שלמים בגודל n כך שלכל $1 < i \le n$ מתקיים $\left|A[i]-A[i-1]\right| \le 1$. כמו כן נתון ש: $A[n] > A[1]$.

א. (15 נק׳) בהינתן איבר z כך ש: $A[1] \le z \le A[n]$, כתבו אלגוריתם חיפוש יעיל עבורו ונתחו את סיבוכיותו.

(20 נק') נתונה השגרה הבאה שהקלט שלה הוא שני מערכים ממוינים A ו B בגודל n (האינדקס הראשון בשניהם הוא 1, ניתן להניח שהאברים בכל אחד מהמערכים הם ייחודיים, כמו-כן נסמן $i++$ למשל כקיצור להצבה $i \leftarrow i+1$) :

Count_What(A,B,n)

$i \leftarrow 1; j \leftarrow 1; s \leftarrow 0$

While $i \leq n$ and $j \leq n$

If $A[i] \leq A[j]$ then $i++$

Else if $A[i] > A[j]$ then $j++$

Else $s++; i++; j++$

Return s

1. (5 נק') מה מחזירה השגרה כפונקציה של הקלט שלה ? נדרשת תשובה מילולית של שורה אחת קצרה.
2. (10 נק') נסחו שמורת לולאה מתאימה באמצעותה ניתן להוכיח זאת (אין צורך להוכיח את שמורת הלולאה). יש לנסח את השמורה כתנאי מדויק המנוסח באמצעות המשתנים המופיעים בשגרה. נדרשת תשובה של שתי שורות לכל היותר.
3. (5 נק') נתחו בקצרה את סיבוכיות השגרה.

(20 נק') נתונה השגרה הבאה שהקלט שלה הוא שני מערכים ממוינים A ו B בגודל n
(האינדקס הראשון בשניהם הוא 1, ניתן להניח שהאברים בכל אחד מהמערכים הם ייחודיים,
כמו-כן נסמן $i++$ למשל כקיצור להצבה $i \leftarrow i+1$) :

```
Count_What(A,B,n)
    i ← 1; j ← 1; s ← 0
    While i ≤ n and j ≤ n
        If A[i] ≤ A[j] then i++
        Else if A[i] > A[j] then j++
            Else s++; i++; j++
    Return s
```

כתוב אלגוריתם המקבל כקלט מערך (לא ממויין),
ומחזיר את ההפרש המינמלי בין שני איברים כלשהם במערך.
הוכח את האלגוריתם.

סיבוכיות

ב. (8 נק׳) קבעו את היחס האסימפטוטי בין $(\log n)!$ לבין n^k עבור k קבוע חיובי כלשהו.

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

*פתרו ע"י עץ רקורסיה

שאלה ממבחן-

פתרו את נוסחת הנסיגה $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{6}\right) + n$

***נסו לחסום מלמעלה ומלמטה ע"י נוסחאות נסיגה פשוטות**

שאלה 1

(16 נקי) **א.** תהאנה $f, g: \mathbb{N} \to \mathbb{N}$ פונקציות עולות. הוכיחו או הפריכו כל אחת מהטענות הבאות.

1. $f(n) = \Omega(\max(f(n), g(n)))$ או $g(n) = \Omega(\max(f(n), g(n)))$.

א. נתחו את סיבוכיות הזמן של Alg1 כתלות ב- n.

```
Alg1 (n)
    k ← n
    while k > 0
        k ← ⌊k/5⌋
        for j ← 1 to ⌊log n⌋
            for i ← 1 to ⌊n/2⌋
                print(i)
```

ב. נתחו את סיבוכיות הזמן של Alg2 כתלות ב- n ו- m.
הניחו כי חישוב ביטוי מהצורה a^b דורש זמן $\Theta(b)$.

```
Alg2 (n, m)
    while n > 0
        a ← m^n
        i ← 1
        while i ≤ a
            i ← i * 2
        n ← n - 1
```

(20 נק') נתונה השגרה הבאה שהקלט שלה הוא שני מערכים ממוינים A ו B בגודל n (האינדקס הראשון בשניהם הוא 1, ניתן להניח שהאברים בכל אחד מהמערכים הם ייחודיים, כמו-כן נסמן $i++$ למשל כקיצור להצבה $i \leftarrow i+1$) :

```
Count_What(A,B,n)
    i ← 1; j ← 1; s ← 0
    While i ≤ n and j ≤ n
        If A[i] ≤ A[j] then i++
        Else if A[i] > A[j] then j++
            Else s++; i++; j++
    Return s
```

1. (5 נק') מה מחזירה השגרה כפונקציה של הקלט שלה ? נדרשת תשובה מילולית של שורה אחת קצרה.
2. (10 נק') נסחו שמורת לולאה מתאימה באמצעותה ניתן להוכיח זאת (אין צורך להוכיח את שמורת הלולאה). יש לנסח את השמורה כתנאי מדויק המנוסח באמצעות המשתנים המופיעים בשגרה. נדרשת תשובה של שתי שורות לכל היותר.
3. (5 נק') נתחו בקצרה את סיבוכיות השגרה.

שאלה 3

הוכח או הפרך – אם נתון ש $f(n) = \omega(g(n))$ וגם $g(n) = \omega(h(n))$ אזי $f(n) = \omega(h(n))$

מהו היחס האסימפטוטי בין הפונקציות

$$n^{\frac{\lg\lg n}{\lg n}} \qquad 3^{\sqrt{n}}$$

ב. (8 נק') קבעו את היחס האסימפטוטי בין $8^{\log 4n}$ לבין n^k עבור k קבוע חיובי כלשהו.

ג. (8 נק') האם מתקיים $f(n)=O\big(f(n)+k\big)$ לכל f חיובית ולכל k טבעי?

הוכיחו או הפריכו כל אחת מהטענות הבאות באופן פורמלי, תוך שימוש בהגדרות לחסמים אסימפטוטיים:

א. $\log(n\text{-}\log n) = \Theta(\log n)$

ב. $2^{2\log n} = \Theta(n\log n)$

ג. $2^{2n+1} + 3^n = O(2^{2n})$

ד. $\binom{n}{3} = \Theta(n^3)$

הוכח או הפרך

אם $f(n) = O(g(n))$ אזי $g(n) = O(f(n))$

נתון מערך A בגודל n. ידוע ש- $n - \lceil \sqrt{n} \rceil$ האיברים הראשונים שלו ממוינים.

הציעו אלגוריתם שממיין את A בסיבוכיות זמן ליניארית.

נוסחאות נסיגה

נתונה נוסחת הנסיגה $T(n)=2T(n/2) + f(n)$.
תנו דוגמה לפונקציה $f(n)$ שעבורה מתקיים המקרה השלישי של משפט האב, אבל לא מתקיים תנאי הרגולריות של מקרה זה.
רמז: בפונקציה תהיה הפרדה למשל בין n זוגי ואי-זוגי.

פתרו את נוסחת הנסיגה

$$T(n) = T(n-1) + \frac{n}{\log n}$$

א. נתון האלגוריתם הבא: הקלט הוא מספר טבעי n ומערך A בן n מספרים שלמים. ראשית האלגוריתם בודק האם $n = 1$ ואם כן אז הוא עוצר. אחרת, עבור $n > 1$, האלגוריתם מבצע n פעמים מיון הכנסה על החצי השמאלי של מערך הקלט וכן מבצע 30 פעמים מיון-מיזוג על החצי הימני של מערך הקלט, לאחר מכן הוא קורא לעצמו ברקורסיה על החצי השמאלי בלבד. יש לנתח את סיבוכיות האלגוריתם באמצעות ניסוח נוסחת נסיגה מתאימה (מנומקת כהלכה) ופתרונה.

שנו את אלגוריתם החיפוש הבינארי הרקורסיבי כך שבמקום לחלק את המערך לשני חלקים (כמעט) שווים, הוא יחלק אותו ביחס 1:2, כלומר: לחלק אחד בגודל שליש וחלק אחר בגודל שני שלישים. כיתבו את האלגוריתם החדש בפסאודו-קוד ונתחו את סיבוכיותו.

מצאו פתרון אסימפטוטי הדוק עבור נוסחת הנסיגה הבאה:

$$\begin{cases} T(1) = c > 0 \\ T(n) = 16T(n/4) + n^{\alpha} \cdot \lg^{\alpha+1} n \end{cases}$$

α הוא פרמטר ממשי חיובי.

פתרו את נוסחת הנסיגה:

$$T(n) = T(n-1) + n^2 \lg n + n^2$$

ערימה

תארו מבנה נתונים המאפשר ביצוע הפעולות הבאות בסיבוכיות הנדרשת:

- Init(S) – אתחול המבנה, בהינתן סדרת איברים S באורך m, בזמן $O(m)$.
- Insert(x) – הוספת x למבנה, בזמן $O(\log n)$ (n הוא מספר האיברים הנוכחי).
- Find-min – החזרת המינימום, בזמן $O(1)$.
- Find-max – החזרת המקסימום, בזמן $O(1)$.
- Del-min – הוצאת המינימום מהמבנה, בזמן $O(\log n)$.
- Del-max – הוצאת המקסימום מהמבנה, בזמן $O(\log n)$.

במכללה מסוימת מלמדים n מרצים. נתוני המרצים (שם, כתובת, שכר) שמורים בקובץ. אנו מעוניינים להדפיס את שמותיהם של m המרצים שמרוויחים את השכר הגבוה ביותר.

<u>תנו פתרון יעיל תחת ההגבלות הבאות:</u>

- סיבוכיות הזיכרון הנוסף המותרת: $O(m)$. נתון $m=o(n)$.
- מותר לעבור על נתוני הקובץ פעם אחת בלבד.

תארו אלג' יעיל ($O(nlogk)$) למיזוג k רשימות ממוינות לרשימה ממוינת אחת בת n איברים.

נתאר אלגוריתם מיון עבור מערך A בגודל n:

- מחלקים את A ל- $\lceil\sqrt{n}\rceil$ קטעים, כל אחד בגודל $\lfloor\sqrt{n}\rfloor$ לפחות; לדוגמה, עבור $n=8$ נחלק את המערך ל-3 חלקים בגדלים 2,3,3 (עקרונית ניתן גם 2,2,4 אך דמיינו חלוקה שווה ככל הניתן).
- על כל אחד מ- $\lceil\sqrt{n}\rceil$ הקטעים, קוראים לאלגוריתם שלנו באופן רקורסיבי.
- ממזגים את $\lceil\sqrt{n}\rceil$ הקטעים הממוינים.

(15 נקי) א. תארו אלגוריתם למיזוג הקטעים בזמן $O(n\cdot\lg n)$; אין להשתמש בשגרת מיון.

(5 נקי) ב. כתבו את נוסחת הנסיגה לחישוב זמן הריצה של האלגוריתם.

(10 נקי) ג. פתרו את נוסחת הנסיגה.

הערה: מותר בכל שלב לרשום $\sqrt{n}$ במקום $\lceil\sqrt{n}\rceil$ או $\lfloor\sqrt{n}\rfloor$.

שאלה 2

נתונה סדרה של k ערמות מינימום H_i, $i=0,1,...,k-1$, כאשר גודל הערמה H_i הוא 2^i לכל $i=0,1,...,k-1$. נניח שכל הערמות ממומשות בעצים בינריים, בעזרת מצביעים.

א' (10 נק') הראו כיצד ניתן למזג את כל הערמות לתוך ערמה אחת H בגודל 2^k-1, בזמן ריצה $O(k^2)$.

ב' (10 נק') הראו כיצד ניתן לפרק את הערמה H בגודל 2^k-1 לסדרת הערמות H_i, $i=0,1,...,k-1$, בזמן ריצה $O(k^2)$.

ג' (5 נק') מהם זמני הריצה בשני הסעיפים הקודמים בהנחה שכל הערמות ממומשות במערכים?הסבירו את תשובתכם.

שאלה 3

תהי A ערימת **מינימום** עם n איברים שונים זה מזה.

א. (12 נק') כתבו אלגוריתם למימוש הפעולה *IncreaseKey(i,val)* המגדילה את ערכו של A[i] לערך *val* (ניתן להניח ש-*val* גדול מהערך ב-A[i] וכן ניתן להניח שגם לאחר ביצוע הפעולה כל האיברים שונים זה מזה).

ב. (13 נק') נתון הפסאודו קוד הבא:

```
MaxVal ← Max(A)+1
for i←n downto 1 do
        IncreaseKey(i,MaxVal)
        MaxVal← MaxVal+1
```

כאשר *Max()* היא שגרה למציאת מקסימום במערך.

הראו שזמן הריצה של הפסאודו קוד שלעיל הוא ליניארי.

שאלה 2

נתון מערך $A[1,...,n]$ של n אברים, יש לבצע עבודת הכנה בסיבוכיות זמן של $O(n)$ וכן $O(n)$ זכרון חיצוני כך שלאחריה ניתן יהיה לענות על כל שאילתא $min(i,j)$ עבור $1 \leq i \leq j \leq n$ שמחזירה את האבר המינימלי בתת-מערך $A[i,...,j]$ בזמן $O(lgn)$. האלגוריתם צריך להיות מתואר במדויק (אפשר באופן מילולי) ויש להסביר באופן כללי מדוע האלגוריתם פועל כנדרש אך אין צורך בהוכחה פורמלית מדויקת. הדרכה : בנו ערמה שבסיסה הוא מערך הקלט A. ניתן להניח ש $n = 2^k$.

תרגיל

הכניסו לערימה שני ערכים – a ואחריו b, ואז מחקו אותם מהערימה בסדר שבו הם הוכנסו, כלומר קודם מחקו את a ואז את b. האם הערימה תחזור בהכרח להיות בדיוק כמו שהיתה לפני ההכנסה?

מיון מהיר

2. הדגימו את ריצתו של מיון מהיר על מערך ממוין, שכל איבריו שונים זה מזה. מהי סיבוכיות זמן הריצה במקרה זה?

1. נשנה את שורה 1 באלגוריתם Quick-Sort ל:

1. **if** $r\text{-}p \geq k$

כלומר, לא ממיינים תתי-מערכים בגודל k או פחות.

במקרה כזה אומרים שהפלט של Quick-Sort הוא מערך "כמעט ממוין עם שגיאה בגודל k".

נתחו את סיבוכיות הזמן של אלגוריתם זה במקרה הגרוע ובמקרה הטוב, כתלות ב- n וב- k.

2. הדגימו את ריצתו של מיון מהיר על מערך $A[1..n]$ הממוין בסדר הפוך, שכל איבריו שונים זה מזה. מהי סיבוכיות זמן הריצה במקרה זה?

*מיון מהיר ומיון הכנסה מאוד מושפעים מהתפלגות הקלט. ולכן יש הרבה שאלות בסגנון הרצה של מיונים אלה על קלט מסויים.
מיון מיזוג, לשם ההשוואה, לא מושפע מהפלגות הקלט.

א. (10 נק') נתחו את זמן הריצה של מיון-מהיר (לא אקראי) על המערך הבא כפונקציה של n (עבור $n \geq 5$) : $A=[2,1,3,\ldots,n-2\ ,n,n-1]$. כלומר A התקבל מהמערך הממוין $[1,\ldots,n]$ שהוחלפו בו שני המקומות הראשונים ושני המקומות האחרונים, למשל עבור $n=8$: $A=[2,1,3,4,5,6,8,7]$.

שאלה 3

א. הסבירו את התכונה הבאה של שגרת החלוקה : סדרת האברים הקטנים או שווים לאבר הציר בקלט שומרת על הסדר היחסי בין אבריה גם בפלט, כלומר : אם a ו b שניהם קטנים או שווים לאבר הציר ו a נמצא לפני b במערך הקלט אז בפלט של שגרת החלוקה a שוב יימצא לפני b.

ב. האם הטענה מסעיף א תקפה גם עבור סדרת האברים בקלט שגדולים ממש מאבר הציר ? אם כן הסבירו זאת במדויק, אם לא : תנו דוגמה נגדית מתאימה (פשוטה ככל הניתן).

ג. נתחו את סיבוכיות זמן הריצה של מיון-מהיר על המערך הבא :

$$[m+1,\ldots,n,1,\ldots,m]$$

כאשר $m > \frac{n}{2}$. למשל אם m=6 ו n=10 אז המערך יהיה שווה ל $[7,8,9,10,1,2,3,4,5,6]$

ניתן להשתמש בטענה מסעיף א (גם אם לא הוכחתם אותה).

חציונים וערכי מיקום

1. נגדיר <u>איבר רוב</u> במערך בגודל n כאיבר שמופיע יותר מ- $n/2$ פעמים.

   הציעו אלגוריתם, שבהינתן מערך בגודל n מוצא איבר רוב, אם קיים כזה, ואחרת מודיע שלא קיים איבר רוב.

מקרה פרטי: אם נתון כי מערך הקלט ממוין. האם ניתן לשפר את זמן הריצה?

2. נתון מערך A בגודל n של איברים כלשהם.

א. הראו אלגוריתם להדפסת $\lfloor\sqrt{n}\rfloor$ האיברים הקטנים במערך בסדר ממוין, בזמן ליניארי במקרה הגרוע. נמקו מדוע לא ניתן לפתור את הבעיה בזמן $o(n)$.

ב. הוכיחו כי לא ניתן להדפיס את $\lfloor n/10\rfloor$ האיברים הקטנים במערך בסדר ממוין, בזמן $o(n\log n)$.

3. נתון מערך ובו m מספרים שונים זה מזה כלשהם.
הציעו מבנה נתונים לביצוע הפעולות הבאות, תוך עמידה בדרישות סיבוכיות הזמן:

- Init — אתחול מבנה הנתונים ב- $O(m)$.
- Insert(x) — הוספת x למבנה ב- $O(\log n)$, n מספר האיברים במבנה בעת ביצוע הפעולה.
- Find-Mid — הדפסת ערך החציון ב- $O(1)$.
- Del-Mid — הוצאת החציון מהמבנה ב- $O(\log n)$.

תארו תחילה מה מכיל המימוש שלכם, ואח"כ הסבירו כיצד מתבצעת כל פעולה ומדוע היא עומדת בדרישות הסיבוכיות.

שאלה 2 (25 נק׳)

עבור מספרים טבעיים a ו b המיוצגים בבסיס עשרוני נגיד ש a חשוב יותר מ b אם סכום הספרות של a גדול או שווה מסכום הספרות של b. נתון מערך A בן n מספרים טבעיים, הציעו אלגוריתם המקבל כקלט את A ומספר טבעי k (בין 1 ל- n) ומדפיס את k האברים החשובים ביותר ב A. יש לנתח את נכונותו וסיבוכיותו של האלגוריתם עבור שני המקרים הבאים:

א. (15 נק׳) לא נדרש שהאברים יודפסו בסדר ממוין. במקרה זה על האלגוריתם לרוץ בזמן $\Theta(n)$ במקרה הגרוע.

ב. (10 נק) נדרש שהאברים יודפסו בסדר ממוין וידוע ש $k = \Theta(\sqrt{n})$.

- נתון מערך A בעל n איברים ומספר שלם k >= n.
- כתבו אלגוריתם שזמן ריצתו $O(n)$ שמוצא את k האיברים הקרובים ביותר לחציון של A.

0	1	2	3	4	5	6	7	8	9
5	6	3	10	8	9	4	2	1	7

אם $k = 4$ צריכים למצוא את 4 המספרים הקרובים ביותר ל-5.

- נתונים שני מערכים *ממוינים* A ו-B בגודל n כל אחד. מצאו אלגוריתם המחזיר את החציון של כל $2n$ האיברים בזמן לוגריתמי. (תרגיל 9.3-8)

שאלה 1 (25 נקודות)

נזכיר – אלגוריתם אופטימלי הוא אלגוריתם שזמן ריצתו שווה בסדר גודל לחסם התחתון של הבעיה. באופן לא פורמלי, אלגוריתם אופטימלי הוא האלגוריתם "הכי טוב שיש" לבעיה.

נתון אלגוריתם מיון M הפועל באופן הבא:

בהינתן מערך A באורך n

- מוצאים את החציון שלו (בעזרת אלגוריתם אופטימלי);
- מבצעים חלוקה סביב החציון (בעזרת אלגוריתם אופטימלי);
- ממיינים את החלק השמאלי (בעזרת אלגוריתם אופטימלי);
- מפעילים את האלגוריתם M על החלק הימני של החלוקה באופן רקורסיבי.

(5 נק') א. מהם האלגוריתמים האופטימליים בכל שלב?

(5 נק') ב. הוכיחו שהאלגוריתם M ממיין נכון את המערך.

(8 נק') ג. כתבו נוסחת נסיגה עבור האלגוריתם M ופתרו אותה.

(7 נק') ד. נבצע את השינוי הבא באלגוריתם: במקום למצוא חציון נמצא את ערך המיקום ה-$n/3$, נבצע את החלוקה סביבו ונמשיך כאמור לעיל. כתבו הפעם נוסחת נסיגה עבור האלגוריתם M ופתרו אותה.

נתונה סדרה S בת n מספרים.

א. הוכיחו את הטענה: ב- S קיימים לכל היותר שלושה מספרים החוזרים על עצמם יותר מ- $\lfloor n/4 \rfloor$ פעמים.

ב. כתבו אלגוריתם למציאת כל האיברים המופיעים בסדרה יותר מ- $\lfloor n/4 \rfloor$ פעמים. זמן הריצה הנדרש הוא $\Theta(n)$.

ג. יהי k מספר טבעי גדול מ-4. כתבו אלגוריתם למציאת כל האיברים המופיעים בסדרה יותר מ- $\lfloor n/k \rfloor$ פעמים. זמן הריצה הנדרש הוא $\Theta(n \log k)$.

חסם תחתון לבעיית המיון

2. פרופסור כלשהו במחלקה כלשהי במכללה כלשהי בצפון הארץ טוען שלאחר שנות מחקר רבות, מצא אלגוריתם ליניארי שמקבל ערימה בת n איברים, ומדפיס את איבריה ממוינים.

האם הייתם נותנים לפרופסור ללמד מבני נתונים? אם כן – הראו אלגוריתם כמו זה שמציע הפרופסור. אם לא – הוכיחו כי טענתו לא יכולה להיות נכונה.

2. נתון מערך A בגודל n של איברים כלשהם.

א. הראו אלגוריתם להדפסת $\lfloor\sqrt{n}\rfloor$ האיברים הקטנים במערך בסדר ממוין, בזמן ליניארי במקרה הגרוע. נמקו מדוע לא ניתן לפתור את הבעיה בזמן $o(n)$.

ב. הוכיחו כי לא ניתן להדפיס את $\lfloor n/10\rfloor$ האיברים הקטנים במערך בסדר ממוין, בזמן $o(n\log n)$.

נתון מערך A של מספרים שלמים בגודל n כך שלכל $1<i\le n$ מתקיים $|A[i]-A[i-1]|\le 1$. כמו כן נתון ש: $A[n]>A[1]$.

א. (15 נק') בהינתן איבר z כך ש: $A[1]\le z\le A[n]$, כתבו אלגוריתם חיפוש יעיל עבורו ונתחו את סיבוכיותו.

ב. (10 נק') הוכיחו שהאלגוריתם שכתבתם אופטימלי (כלומר שאין אלגוריתם בסיבוכיות זמן ומקום טובים יותר).

5. <u>בעיית הוקטור הפרבולי</u>

בבעיית הוקטור הפרבולי נתון מערך בגודל n של שלמים שונים (אין חסם על גודלם) וצריך לסדר מחדש את איברי המערך כך שיתקיים:

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

מערך שמסודר באופן הנ"ל נקרא וקטור פרבולי.

לדוגמה, המערך הבא הינו וקטור פרבולי: $< 3, 4, 12, 9, 8, 6 >$

א. תנו חסם תחתון (במונחים של Ω) לסיבוכיות הזמן הדרושה לפתרון בעיית הוקטור הפרבולי.

ב. הציגו אלגוריתם אופטימלי (הן מבחינת סיבוכיות <u>זמן</u> והן מבחינת סיבוכיות <u>מקום</u>) לפתרון בעיית הוקטור הפרבולי.

הוכיחו שכל אלגוריתם מבוסס השוואות הפותר את בעיית החזרת k האיברים הקטנים ביותר בסדר ממוין (מתוך סדרת n איברים נתונים), רץ בזמן $\Omega(k \cdot \lg n)$ במקרה הגרוע.

מיונים בזמן לינארי

נתונות m קבוצות $S_1, S_2, ..., S_m$. כל קבוצה מכילה מספרים שלמים בתחום $1..n$.

נתון כי $\sum_{i=1}^{m} |S_i| = O(n)$ ($|S_i|$ הוא מספר האיברים בקבוצה S_i).

הציעו אלגוריתם הממיין את כל הקבוצות S_i $i = 1, 2, ..., m$.

כלומר, האלגוריתם צריך להחזיר m קבוצות ממוינות.

סיבוכיות הזמן הנדרשת: $O(n)$

3. **<u>פלינדרום</u>** (Palindrome) באורך n הוא סדרת תווים $a_1,a_2,\ldots,a_n$ המקיימת $a_i = a_j$ לכל $i = n-j+1$, עבור $1 \le i, j \le n$.

נתונה סדרה באורך n של מספרים שלמים בין 1 ל- $3n$, וידוע שכל מספר מופיע כמות זוגית של פעמים. האם ניתן להפוך את הסדרה הנתונה לפלינדרום באורך n, שמכיל את אותם מספרים בדיוק, בזמן ריצה $o(n\log n)$?

הראו כיצד או הוכיחו כי לא ניתן.

8.2-4

תארו אלגוריתם אשר בהינתן n מספרים שלמים בתחום 0 עד k, מבצע עיבוד מקדים על הקלט שלו, ואז עונה בתוך זמן $O(1)$ על כל שאלה מהצורה "כמה מ-n השלמים שייכים לתחום $[a..b]$?". האלגוריתם צריך לבצע את העיבוד המקדים בזמן $\Theta(n + k)$.

נגדיר **מחלקת שקילות** של מילים בשפה האנגלית באופן הבא:

שתי מילים שייכות לאותה מחלקת שקילות אם הן מורכבות בדיוק מאותן אותיות.

למשל, המילים read, dear, dare שייכות כולן לאותה מחלקת שקילות.

כתבו אלגוריתם המקבל רשימה באורך n של מילים בנות ארבע אותיות באנגלית, ומוצא את מחלקת השקילות הגדולה ביותר. סיבוכיות זמן הריצה של האלגוריתם צריכה להיות $O(n)$.

4. נתונים n מספרים שלמים בתחום $[0, n^2-1]$. הציעו דרך יעילה למיינם.

רמז: מהו המספר המקסימלי בן שתי ספרות בבסיס n?

נתונים n קטעים $[a_i,b_i]$, $i=1,\dots,n$, בתוך הקטע $[0,n^d]$ על הישר הממשי (d קבוע שלם

חיובי, a_i,b_i שלמים, $0\le a_i\le b_i\le n^d$, לכל $i=1,\dots,n$).

כתבו אלגוריתם למציאת שלם z השייך למספר מכסימלי של קטעים $[a_i,b_i]$. זמן הריצה הנדרש

שלו **ליניארי**.

נתונה רשימה של n קטעים $[a_i,b_i]$, $i=1,\ldots,n$ על הישר הממשי; כלומר, נתונה רשימה של n זוגות של מספרים ממשיים (a_i,b_i), $i=1,\ldots,n$, המקיימים את התנאי $a_i<b_i$ לכל $i=1,\ldots,n$.

כתבו אלגוריתם למציאת קטע $[a_i,b_i]$, המקיים את התנאי: מספר הקטעים $[a_l,b_l]$ שעבורם $b_l<a_i$ שווה למספר הקטעים $[a_r,b_r]$ שעבורם $b_i<a_r$; כלומר, מספר הקטעים שנמצאים משמאל לקטע $[a_i,b_i]$ שווה למספר הקטעים שנמצאים מימינו. זמן הריצה הנדרש של האלגוריתם: $O(n\cdot\lg n)$ במקרה הגרוע. אם קיים קטע כזה, האלגוריתם יחזיר את האינדקס i; אחרת, הוא יחזיר NIL.

נתון מערך P באורך $m+n$ של מספרים שלמים. ידוע ש-m המספרים הראשונים שייכים לתחום $\left[1..n^2\right]$ ו-n המספרים האחרונים שייכים לתחום $\left[1..m^3\right]$.

תארו אלגוריתם למיון המערך P בזמן $O(m+n)$.

שאלה 5

נתון מערך $A[1..n]$, ממוין, וערך כלשהו z.

כתבו אלגוריתם הקובע כמה פעמים מופיע הערך z במערך.

נסו להגיע לזמן ביצוע $O(\lg n)$ במקרה הגרוע.

שאלה 6

נתון מערך A של מספרים שלמים, באורך בלתי-מוגבל. n האיברים הראשונים ממויינים בסדר עולה (לא יורד); כל האיברים האחרים מכילים את הערך ∞.

כתבו אלגוריתם הפותר את בעיית חיפוש ערך סופי כלשהו z (ז"א z שונה מ- ∞) בין n האיברים הראשונים (יוחזר האינדקס של אחד האיברים שערכם z, או הערך 0 אם z לא נמצא במערך); זמן הריצה של האלגוריתם חייב להיות $O(\lg n)$.

הערה: n הוא משתנה שערכו **אינו ידוע מראש**.

נתונה רשימה דו-מקושרת L שאורכה בלתי-ידוע. אין לנו מצביעים לראשה ולזנבה של הרשימה, אבל יש לנו מצביע לאיבר מסוים s. מחפשים איבר x השייך ל- L שמיקומו ברשימה לא ידוע.

(10 נק') א. מבצעים את שגרת החיפוש הבאה:

לכל $k=1,2,\ldots$ עוברים ברשימה k צעדים שמאלה וחזרה, אחר כך k צעדים ימינה וחזרה, עד שמוצאים את x. נסמן ב- n את המרחק של x מ- s.

הראו שהשגרה רצה בזמן $\Theta(n^2)$ במקרה הגרוע.

(10 נק') ב. נסו לשפר את שגרת החיפוש כך שזמן ריצתה יהיה $\Theta(n)$.

1. שאלה 10.1-6 מספר הלימוד.
הראו כיצד ניתן לממש תור באמצעות שתי מחסניות. נתחו את זמן הריצה של הפעולות על התור.

*ממשו את התור, כך שזמן הריצה של הפעולות על פני הכנסה והוצאה של n איברים יהיה $\theta(n)$.

2. נגדיר "מחסנית מינימום" כ- ADT התומך בפעולות הבאות:

- Create - אתחול מבנה הנתונים כאשר הוא ריק.
- Insert(x) - הכנסת המספר x למבנה.
- RemoveLast - הוצאת המספר שהוכנס אחרון והחזרתו כפלט.
- Min - החזרת המספר הקטן ביותר במבנה (ללא הוצאתו).
- ChangeMin(k) - שינוי ערך המספר הקטן ביותר במבנה ל- k.

מותר להניח כי בכל זמן נתון כל המספרים במבנה שונים זה מזה.

א. הציעו מימוש ל"מחסנית מינימום", כאשר סיבוכיות הזמן הנדרשת לארבע הפעולות הראשונות היא $O(1)$, ולפעולה ChangeMin $O(t)$, כש- t הינו מספר האיברים במבנה שהוכנסו אחרי האיבר המינימלי.

ב. הוחלט להוסיף את הפעולה הבאה:

- Add(d) - הוספת d לכל המספרים במבנה. סיבוכיות זמן נדרשת: $O(1)$.

הסבירו כיצד לממש את הפעולה החדשה, ומהם השינויים הדרושים במימוש הפעולות מסעיף א' כך שלא יהיה שינוי בסיבוכיותן.

שאלה 2

בהנתן מערך $A[1, \dots, n]$, ומספר שלם $1 \le k \le n$, כִּתְבוּ אלגוריתם הבונה מערך $B[1, \dots, n-k+1]$ המקיים כי $B[j] = \max\{A[j], A[j+1], \dots, A[j+k-1]\}$. זמן הריצה של האלגוריתם הוא $\Theta(n\text{Log}(n))$

שאלה 2

בהנתן מערך $A[1, \dots, n]$, ומספר שלם $1 \le k \le n$, כִּתְבוּ אלגוריתם הבונה מערך $B[1, \dots, n-k+1]$ המקיים כי $B[j] = \max\{A[j], A[j+1], \dots, A[j+k-1]\}$. זמן הריצה של האלגוריתם הוא $\Theta(n)$.

ממשו תור באמצעות ערימה.
יעילות פעולות התור צריכה להיות $\theta(\mathrm{Log}(n))$

<u>שאלה</u>

הציעו מבנה נתונים התומך בפעולות הבאות:

א) add(x)- מוסיף איבר לסוף המבנה.

ב) delete(x)- מוציא איבר מההתחלה

ג) mid(x)- מחזיר (בלי להוציא) את האיבר האמצעי במבנה

שאלה

בהינתן מחרוזת S המכילה סוגריים פותחים וסוגרים,

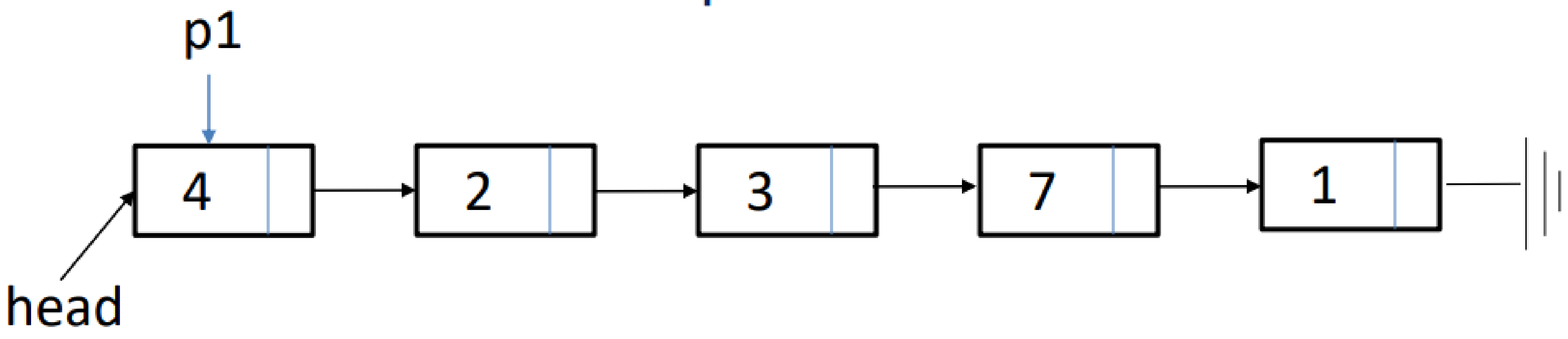
הצע אלגוריתם הבודק את תקינות הסוגריים.

כלומר האלגוריתם בודק כי לכל סוגר פותח-"(", קיים סוגר-")" הסוגר אותו.

דוגמאות תקינות: "()ליע(())", "(((1+x)))" "((()*8()נענ()))"

דוגמאות שאינן תקינות: "יכעכ)(", "(()))"

כתבו אלגוריתם המוצא את האיבר ה-k מסוף הרשימה.



*הניחו שn לא ידוע

סעיף א (15 נק')

נתון מערך P של n מספרים חיוביים. ברצוננו לבנות מערך S בגודל n לפי ההגדרה:

$$S[i] = \max\{k: \ j = i-k+1,\ldots,i \ \text{לכל} \ P[j] \le P[i] \ \text{וגם} \ k \le i\}$$

קרי: המקסימום מבין כל ה-k כך ש k<=i וגם $P[j] \le P[i]$ לכל j בין $i-k+1$ לבין i.

בצורה לא פורמלית: בכל אינדקס i ב-S יהיה המרחק המקסימלי שניתן "להסתכל אחורה" מאינדקס i ב-P ולראות רק איברים שקטנים שווים ממני. לדוגמה, עבור P=[2,7,4,5,9,12], S=[1,2,1,2,5,6]. וודאו תחילה שאתם מבינים את ההגדרה.

כתבו אלגוריתם לבניית המערך S בזמן $O(n)$. **רמז:** השתמשו במחסנית כמבנה עזר.

- תרגיל 10.2-7

כתבו שגרה לא רקורסיבית שזמן ריצתה $O(n)$, ההופכת את סדר האיברים ברשימה חד-מקושרת בת n איברים. השגרה יכולה להשתמש לכל היותר בכמות קבועה של זיכרון, בנוסף למקום הנדרש לאחסון הרשימה עצמה.

עצי חיפוש בינאריים

4. הוכיחו כי לא קיים אלגוריתם לבניית עץ חיפוש בינארי מרשימה נתונה של n איברים, שזמן ריצתו $o(n\log n)$.

נתון אלגוריתם הבונה עח"ב מתוך מערך A ע"י n פעולות הכנסה.
חשבו את יעילותו במקרה הטוב, הרע והממוצע

5. <u>ואריאציה של תרגיל 12.2-7 מספר הלימוד</u>

ניתן לממש סריקה תוכית (in-order) של עץ חיפוש בינארי בעל n צמתים ע"י מציאת המינימום, ואח"כ מציאת העוקב של הצומת הנוכחי n-1 פעמים.

להלן חישוב <u>חסם הדוק שגוי</u> לזמן הריצה במקרה הגרוע:

- גובה העץ הוא $h=\Theta(n)$.
- כל אחת מפעולות Tree-Successor רצה בזמן $\Theta(h)$.
- לכן סה"כ זמן הריצה הוא $(n\text{-}1)\Theta(h) = \Theta(n^2)$.

הסבירו היכן הטעות ומדוע חישוב זה מתאים לחסם עליון בלבד (רמז: איפה צריך להחליף Θ ב- O?).

הסבירו מדוע למעשה זמן הריצה של אלגוריתם זה הוא $\Theta(n)$.

נתון עץ חיפוש בינארי T.

הציעו שיפור עבורו, כך שיהיה ניתן להוסיף לו את השאילתות הבאות:

K_Successor(p,k): מציאת העוקב הk של האיבר שאליו מצביע p,
זמן ריצה O(k).

K_Maximum(k): הדפסת k האיברים הגדולים ביותר בעץ, בזמן ריצה O(k).

*שימו לב- העץ החדש מכיל גם את הפעולות הרגילות של ע"חב

א. כתבו אלגוריתם רקורסיבי, אשר מקבל מערך ממוין A בגודל n ובונה מאיברי המערך עץ חיפוש בינרי מאוזן (עץ שגובהו $\Theta(\lg n)$).

ב. נתחו את זמן הריצה של האלגוריתם.

ג. ציירו את העץ המתקבל מהפעלת האלגוריתם על המערך הבא:

1	2	3	4	5	6	7	8	9	10
15	17	18	32	38	40	49	75	89	94

נתון עץ חיפוש בינארי T. כתבו אלגוריתם שמקבל את T ושני מספרים a, b כך ש $a < b$ ושניהם נמצאים בעץ. האלגוריתם ידפיס את כל המספרים בתחום $[a, b]$ ביעילות $O(h+k)$ כאשר h הוא גובה העץ ו-k הוא מספר האיברים בתחום.

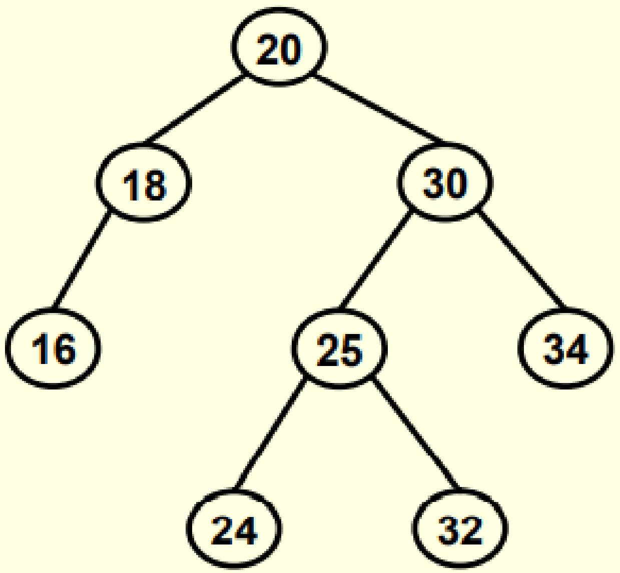
*שימו לב שאתם מבינים שהרעיון הנאיבי של מציאת a, ואז הרצת עוקב() K פעמים,
הוא יקר מידי ויעלה O(kh), ובנוסף יהיה חסום מלמעלה ע"י O(n).
וודאו שאתם מבינים שגם אם ננתח לשיעורין את האלגוריתם הנאיבי הזה, זמן
הריצה לא יהיה ליניארי בk...

פתרון שגוי

כתבו שגרה הבודקת האם עץ בינארי נתון הוא עח"ב.

```
IsBST(node)
if node = null then return true

if (key[node] > key[left[node]] and
    key[node] < key[right[node]] and
    IsBST(left[node]) and
    IsBST(right[node]) then
    return true
else return false
```



בניגוד לערמה, תכונת העח"ב היא לא מקומית. תיקון:

$\boldsymbol{isBST \leftarrow isBST_{left}}$ **and** $\boldsymbol{isBST_{right}}$ **and** $\boldsymbol{max_{left}}$ **<= key[node] <=** $\boldsymbol{min_{right}}$

ניתן לתחזק ערכים אלו במהלך הקריאות הרקורסיביות

כתוב גרסה רקורסיבית לשאילתות
Tree-Maxinum(x),
Tree-Minimum(x)

2. סטודנט הפעיל סיורים *inorder* ו- *preorder* על עץ בינארי שהיה ברשותו. כעבור זמן מה, הוא גילה לתדהמתו שאיבד את העץ, אך יש עדיין בידו את תוצאות הסיורים.

עזרו לסטודנט לשחזר את העץ. מה סיבוכיות הפתרון במקרה הגרוע / הטוב ?

inorder	:	2	6	4	7	1	3	8	5	9	10
preorder	:	1	2	4	6	7	3	5	8	9	10

*שימו לב שלא מדובר בעץ חיפוש

נתונים שלושה עצים בינאריים – A, B, C שכל אחד מהם מכיל את **כל** המספרים n ,…, 1, 2, 3 ללא חזרות. נתון שגם סריקת in order על עץ A וגם סריקת pre order על עץ B וגם סריקת post order על עץ C מפיקות את הפלט

1 2 3 4 … n

עבור כל אחת מהטענות הבאות כתבו אם היא נכונה או לא. נמקו או תנו דוגמא נגדית.

א- A הוא בהכרח עץ חיפוש בינארי.

ב – B הוא בהכרח עץ חיפוש בינארי

ג - תוצאת סריקת post order על B תהיה זהה לתוצאת pre order על C.

ד - יתכן שרק A ו-B זהים ו-C שונה.

ה - יתכן שרק B ו-C זהים ו-A שונה.

נתונים שני עצי חיפוש בינאריים, כל אחד בעל n צמתים.

תארו אלגוריתם יעיל למיזוג שני העצים לעץ אחד. מהו זמן הריצה של האלגוריתם ?

עצים אדומים-שחורים

עץ ערכי מיקום