מספר תעודת זהות:

מספר מחברת:

מספר קורס: 0368.2158

סמסטר ב', מועד ב', תשס"ז

י"ד באב תשס"ז 18/10/2007

מבחן במבני נתונים

פרופ' חיים קפלן, אלעד ורבין, ליאור שפירא

משך המבחן שלוש וחצי שעות **(לא תינתן הארכה)**

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
4. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
5. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות. **תשובה ללא נמוק במקום שנדרש נמוק לא תזכה בנקודות.**
6. סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב O().

בהצלחה!!

שאלה	ציון	ניקוד מקס'
1		20
2		14
3		18
4		20
5		14
6		14
סה"כ		100

מספר מחברת:

<u>שאלה 1 (20 נקודות)</u>

א. נתון מערך בגודל n שבו כל תא מכיל את התו '+' או התו '-'. ידוע כי האיבר הראשון במערך שווה ל-'+' והאחרון שווה ל-'-'. תאר אלגוריתם שרץ בזמן **O(log n)** ומוצא אינדקס j כך ש- A[j]='+' וגם A[j+1]='-'.

דוגמה:

	+	+	_	+	+	+	_	_	+	_
index	0	1	2	3	4	5	6	7	8	9

בדוגמה שלהלן התשובות החוקיות הן j=1, j=5 ו-j=8. (מותר להחזיר איזו מהן שרוצים. לא צריך להחזיר את כולן).

פתרון:

ב. הראה חסם תחתון שמוכיח שלכל אלגוריתם שפותר את השאלה מסעיף א', יש סיבוכיות **$\Omega(\log n)$** ב-worst case.
(שים לב שנדרשת זהירות בטיעון בשל העובדה שעבור קלטים מסויימים יש יותר מתשובה אחת חוקית).

פתרון:

ג. נתון מערך בגודל n המכיל מספרים חיוביים. כל האיברים במערך שונים. כמו כן ידוע שהאיבר הראשון והאחרון במערך הם האיברים הקטנים ביותר. תאר אלגוריתם שרץ בזמן **O(log n)** ומוצא אינדקס j כך ש- $A[j-1]<A[j]$ וגם $A[j+1]<A[j]$. במקרה שיש יותר מתשובה אחת חוקית, ניתן להחזיר את איזו מהן שרוצים.

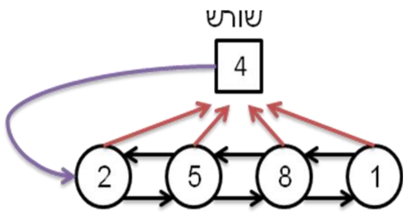
	1	3	4	5	6	7	26	15	14	2
index	0	1	2	3	4	5	6	7	8	9

בדוגמה שלהלן התשובה החוקית היא $j=6$.

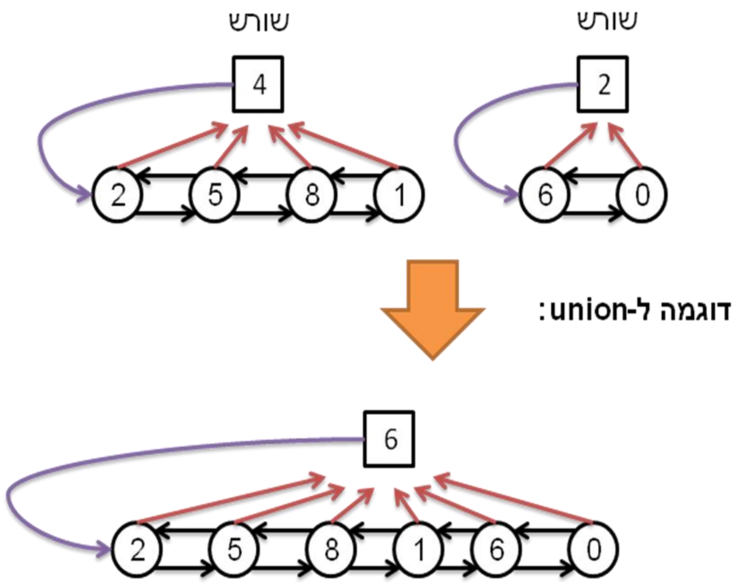
פתרון:

<u>שאלה 2 (14 נקודות)</u>

להלן מבנה נתונים הפותר את בעיית הunion find. נחזיק קבוצה ע"י רשימה משורשרת דו-כיוונית של אבריה כאשר כל אלמנט מצביע לצומת מיוחד המייצג את הקבוצה (נכנה אותו שורש הקבוצה). שורש הקבוצה מצביע לרשימה המשורשרת וגם מכיל את מספר האלמנטים בקבוצה. לדוגמה הנה ייצוג של הקבוצה {2,5,8,1}, גודל הקבוצה (4) נשמר בשורש.



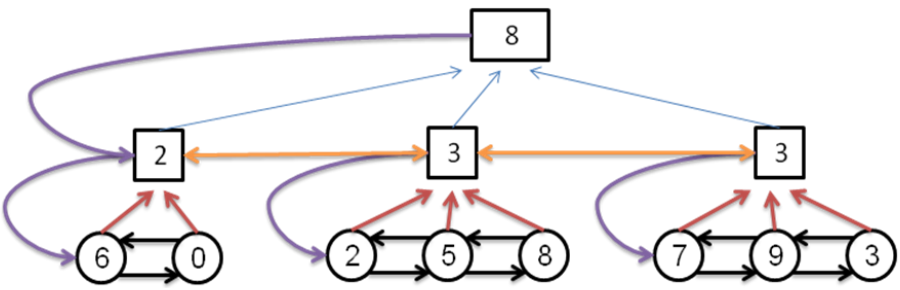
- פעולת ה-find מקבלת מצביע לאלמנט ומחזירה את המצביע לשורש שמאוכסן בצומת המייצג את האלמנט.
- פעולת ה-union מוסיפה את אברי הקבוצה הקטנה בזה אחר זה לרשימה של הקבוצה הגדולה ומעדכנת את המצביע לשורש באברים אלו, שיצביע לשורש בקבוצה הגדולה. בנוסף, הפעולה מעדכנת את מספר האברים בשורש הקבוצה הגדולה, ומוחקת את שורש הקבוצה הקטנה.



א. כמה זמן במקרה הגרוע ייקח לבצע סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק איבר אחד? הנח ש-m>n. הוכח את נכונות תשובתך.

ב. נשנה את מבנה הנתונים באופן הבא. נסמן $k = \lfloor \log n \rfloor$. קבוצות שגודלן k לכל היותר נייצג כמו קודם. קבוצה S בגודל גדול מ-k תיוצג ע"י אוסף של תת קבוצות בגודל k בדיוק (נקרא להן **מלאות**), וקבוצה אחת לכל היותר בגודל קטן מ-k (**חסרה**). תתי קבוצות אלו מיוצגות כמו שתואר בחלק א' של השאלה. שורשי תת הקבוצות מסודרים ברשימה משורשרת דו-כיוונית ומצביעים לצומת חדש המייצג את הקבוצה כולה. שורש זה מכיל מצביע לרשימת השורשים ואת גודל הקבוצה. הקבוצה החסרה (אם קיימת) תהיה תמיד ראשונה ברשימה.

דוגמה לקבוצה גדולה עם שתי קב' מלאות ואחת חסרה, k=logn=3



נבצע את הפעולות באופן הבא:

- נבצע **find** ע"י מציאת שורש תת הקבוצה המאוכסן בצומת המייצג את האיבר ומשם נמשיך לשורש הקבוצה (אם היא קב' גדולה).
- בכדי לבצע פעולת **union** על תת קבוצות, נשתמש בפעולת ה-union כמו שהוגדרה קודם, עם השינוי הבא: במידה וגודל שתי התת קבוצות יחדיו הוא יותר מ-k אזי ה-union יחזיר שתי תת קבוצות, אחת מלאה ואחת חסרה. union זה מתבצע כמו קודם רק שאין הוא מעתיק את כל איברי הקבוצה הקטנה במידה ואין יותר מקום בקבוצה הגדולה.
- **union** של קבוצות נבצע באופן הבא:
  - אם שתי הקבוצות קטנות וגודלן יחדיו קטן או שווה ל-k אזי הפעולה תבוצע כאיחוד תתי קבוצות כמו קודם
  - אם שתי הקבוצות קטנות וגודלן יחדיו גדול מ-k אזי נאחד את תתי הקבוצות ונקבל תת קבוצה אחת מלאה ואחת חסרה. ניצור קבוצה חדשה שאלו הן תתי הקבוצות שלה ונחזיר אותה.
  - אם קב' אחת קטנה ואחת גדולה, נאחד את הקבוצה הקטנה עם תת הקבוצה החסרה של הקבוצה הגדולה. נוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.
  - אם שתי הקבוצות גדולות נעביר את המלאות מן הקבוצה הקטנה יותר לרשימת תת הקבוצות בקב' הגדולה יותר. כמו כן נאחד את החסרות ונוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.

הוכח כי זמן הריצה של סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק אבר אחד הוא $O(n\log\log(n) + m)$.

שאלה 3 (18 נקודות)

נתונה המחרוזת S: ACACACCA

א. ציירו את ה-suffix tree של S.

ב. נגדיר תת מחרוזת z באורך k להיות חזרה מכסימלית ימנית במחרוזת S אם קיימים i ו-j, i≠j כך ש:

$$Z = s_i s_{i+1} \dots s_{i+k-1}$$

$$Z = s_j s_{j+1} \dots s_{j+k-1}$$

$$s_{i+k} \neq s_{j+k}$$

תאר אלגוריתם לינארי המקבל כקלט מחרוזת S ומספר k ומחשב את כל החזרות המכסימליות הימניות באורך k. עבור כל חזרה האלגוריתם מדווח את האינדקס בו מתחיל המופע הראשון שלה.

ג. נתונות אותיות ותדירות הופעתן בטקסט. ציירו את עץ הופמן המתאים לקלט:

אות	א	ב	ג	ד	ה	ו	ז
תדירות	40	100	25	5	100	70	15

מספר תעודת זהות:

מספר מחברת:

<u>שאלה 4 (20 נקודות)</u>

נתונה טבלת hash עם $m=11$ כניסות. כמו כן נתונות שתי פונקציות:

$h_1(key)=key \bmod m$

$h_2(key)=\{key \bmod (m-1)\} + 1$

א. הכניסו את המפתחות {22,1,13,11,24,33,18} לטבלה לפי הסדר (משמאל לימין) בכל אחת מהשיטות הבאות:

א.1. Linear probing כאשר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i) \bmod m$

א.2. Double hashing כאשר h_1 ו-h_2 פונקצית ה-hash פונקצית המרווח. כלומר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i*h_2(k)) \bmod m$

	Linear Probing	Double Hashing
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

ב. מה יקרה אם בא.2 נשתמש ב-h_2 כפונקצית ה-hash וב-h_1 כפונקצית המרווח? כלומר
$h(k,i)=(h_2(k)+i*h_1(k)) \bmod m$

ג. הסבר מדוע כאשר $GCD(h_{step}(k),m)=1$ עבור כל k, התופעה בסעיף ב׳ אינה יכולה להתרחש. הפונקציה $h_{step}()$ היא פונקצית המרווח.

שאלה 5 (14 נקודות)

הוכיחו כי לא יכול להיות אלגוריתם במודל ההשוואות בו עבור קלט באורך n, לפחות חצי מהפרמוטציות האפשריות של המספרים מ-1 עד n ניתנות למיון בזמן ליניארי.

(עבור המחצית השניה של הפרמוטציות, לאלגוריתם מותר להחזיר כל דבר, אפילו סדר שגוי).

שאלה 6 (14 נקודות)

ענו על שני הסעיפים הבאים

א. תארו מבנה נתונים התומך בפעולות הבאות על קבוצה S מתחום סדור מלא

- Quarternian(S): אם יש כרגע ב-S n איברים, מחזיר את ה-$\lceil n/4 \rceil$-י בגודלו
- Average(S): מחזיר את ממוצע האיברים ב-S
- Min(S): מחזיר את האבר הקטן ביותר ב-S
- Max(S): מחזיר את האבר הגדול ביותר ב-S
- Insert(x,S): מוסיף אבר x ל-S
- Delete(x,S) מניחים ש-x נמצא לפני הפעולה ב-S. הפעולה מוציאה את x מ-S

על הפעולות quarternian,average,min,max לקחת $O(1)$ זמן במקרה הגרוע, ועל הפעולות insert ו-delete לקחת $O(\log n)$ זמן במקרה הגרוע. כמו כן תארו בקצרה כיצד לבצע כל פעולה.

ב. שנו במידת הצורך את מבנה הנתונים שהגדרתם בסעיף הקודם כך שנוכל גם למצוא את האלמנט ה-$i + \lceil n/4 \rceil$ בגודלו בזמן $O(\log(i))$. תארו במדויק כיצד לבצע פעולה זו.