

מספר קורס: 0368.2158

סמסטר א', מועד א', תשס"ח

01/05/ 2008

## מבחן במבני נתונים

פרופ' אורי צוויק, פרופ' חיים קפלן, דן פלדמן, ליאור שפירא

משך המבחן שלוש וחצי שעות (לא תינתן הארכה)

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

### הוראות כלליות:

- יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
- המחברות הן לטייטה בלבד ולא תיבדקנה.
- מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
- אין לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
- 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.
- סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב  $O()$ .

## בהצלחה!!

| שאלה | ציון | ניקוד מקסי |
|------|------|------------|
| 1    |      | 14         |
| 2    |      | 14         |
| 3    |      | 15         |
| 4    |      | 15         |
| 5    |      | 14         |
| 6    |      | 14         |
| 7    |      | 14         |
| סה"כ |      | 100        |

**שאלה 1 (14 נקודות)**

שאלה זו מתייחסת למבנה הנתונים union-find שהוצג בכתה. כל קבוצה מיוצגת על ידי עץ כאשר משתמשים ב-union by rank ו-path compression.

א. הוכח/י או הפרך/י: לאחר שבצענו פעולת find על איבר מסוים  $x$ , כל פעולת find נוספת על  $x$  תיקח זמן קבוע (במקרה הגרוע ביותר).

ב. כמה זמן נדרש, במקרה הגרוע ביותר, לביצוע סדרה בת  $m$  פעולות על  $n$  איברים שבה כל פעולות ה-union מבוצעות לפני כל פעולות ה-find? (הסבר/י את תשובתך.)

ג. תאר/י בקצרה כיצד ניתן להוסיף למבנה הנתונים פעולה  $\text{rep}(x)$  שהופכת את האיבר  $x$  לאיבר המייצג בקבוצה שבה הוא נמצא כרגע. מה סיבוכיות הפעולה במקרה הגרוע ביותר ובמקרה ה-amortized? (כאשר קבוצתו של  $x$  מתאחדת עם קבוצה אחרת, האיבר  $x$  יכול להפסיק להיות איבר מיצג.)

**שאלה 2 (14 נקודות)**

אנו מעוניינים בהכללה של מחסנית התומכת בנוסף לפעולות push ו-pop גם בפעולת minimum שמחזירה את האיבר הקטן ביותר במחסנית ובפעולת delete-min המסירה את האיבר הקטן ביותר מהמחסנית.

- א. תאר/י מבנה נתונים התומך בפעולת minimum בזמן קבוע ובכל שאר הפעולות בזמן  $O(\log n)$ , כאשר  $n$  הוא מספר האיברים במחסנית בזמן ביצוע הפעולה. (כל הזמנים כאן הם במקרה הגרוע ביותר.) הוכח/י שהמימוש שתיארת הוא נכון ורץ בזמן הנדרש.

- ב. תאר/י מבנה נתונים, שבמקרה ה-amortized, תומך בפעולות pop ו-delete-min בזמן  $O(\log n)$  ובפעולות push ו-minimum בזמן  $O(1)$ . (כל הזמנים כאן הם amortized.) הוכח/י שהמימוש שתיארת הוא נכון ורץ בזמן הנדרש.

מועד ג'

4 מתוך 16

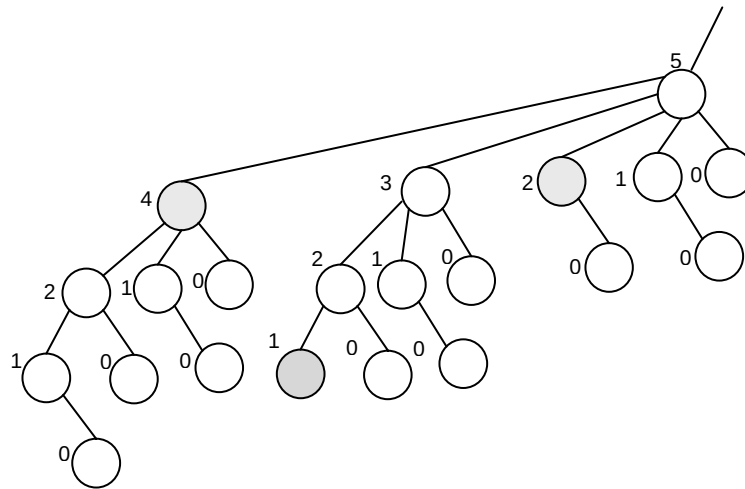
מספר תעודת זהות :

מספר מחברת :

ג. הוכח/י שלא קיים במודל ההשוואות מבנה נתונים שתומך בכל ארבעת הפעולות push, pop, minimum, delete-min בזמן amortized קבוע.

**שאלה 3 (15 נקודות)**

נשנה את ערמת פיבונצ'י באופן הבא. העצים בערמה יהיו עצים בינומיים אולם לכל צומת שאינו שורש יתכן וחסר הבן השמאלי ביותר. כלומר לצומת מדרגה  $r$  יש  $r$  או  $r-1$  בנים, מדרגות  $0, 1, \dots, r-1$  או  $0, 1, \dots, r-2$  בהתאמה. במידה ולצומת מדרגה  $r$  יש רק  $r-1$  ילדים נקרא לו צומת **חסר**. שים לב כי ניתן לדעת אם צומת הוא חסר או לא על ידי השוואת דרגתו לדרגת בנו השמאלי ביותר. אם ההפרש הוא 1, זהו צומת רגיל, אחרת זהו צומת חסר. הנה דוגמא לתת עץ של עץ בינומי חסר, ליד כל צומת רשומה דרגתו, הצמתים הצבועים באפור הם החסרים.



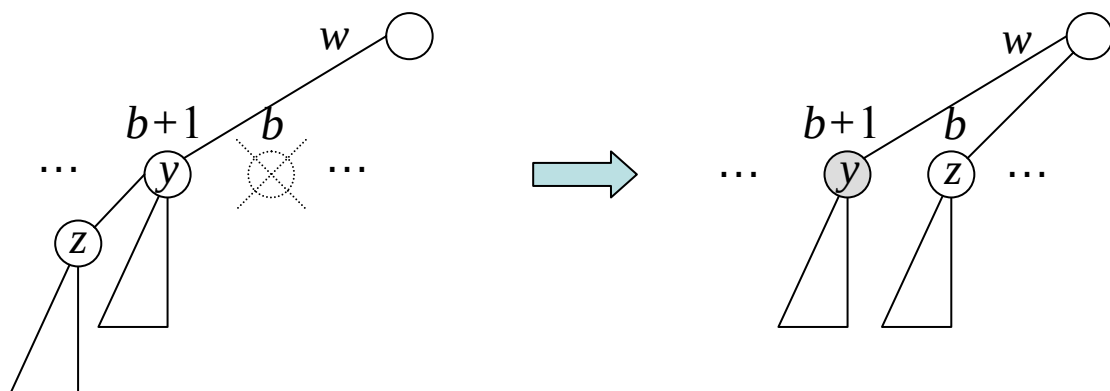
נשנה את הממוש של  $\text{decrease-key}(x, h, \Delta)$  באופן הבא. אם ערכו החדש של  $x$  קטן מזה של אביו, ננתק את  $x$  מאביו ותת העץ ששורשו  $x$  יהפוך לעץ ביער. אם דרגתו של  $x$  היתה  $b$  אזי ל- $w = p[x]$ , אביו של  $x$ , חסר עתה בן מדרגה  $b$ . נתקן זאת על פי אחד מהמקרים הבאים (ראה ציור בעמוד הבא):

- A. ל- $w$  יש בן  $y$  מדרגה  $b+1$  ו- $y$  אינו חסר: יהא  $z$  הבן השמאלי ביותר של  $y$ . הצומת  $z$  הוא מדרגה  $b$ . ננתק אותו מ- $y$ , שיהפוך להיות צומת חסר, ונהפוך אותו לבן מדרגה  $b$  של  $w$ . פעולת התיקון נגמרה.
- B. ל- $w$  יש בן  $y$  מדרגה  $b+1$  ו- $y$  חסר: נהפוך את  $y$  לצומת לא חסר מדרגה  $b$ . ל- $w$  חסר עתה בן מדרגה  $b+1$ . נמשיך בתהליך התיקון עבורו.
- C. ל- $w$  אין בן  $y$  מדרגה  $b+1$  ו- $w$  אינו חסר:  $w$  יהפוך לצומת חסר (מדרגה  $b+1$ ) ופעולת התיקון הסתיימה.
- D. ל- $w$  אין בן  $y$  מדרגה  $b+1$  ו- $w$  חסר:  $w$  יהפוך לצומת לא חסר (מדרגה  $b$ ). אם  $w$  אינו שורש, ננתק את  $w$  מאביו ונמשיך בפעולת התיקון עבור  $p[w]$  שחסר עתה צומת מדרגה  $b+2$ .

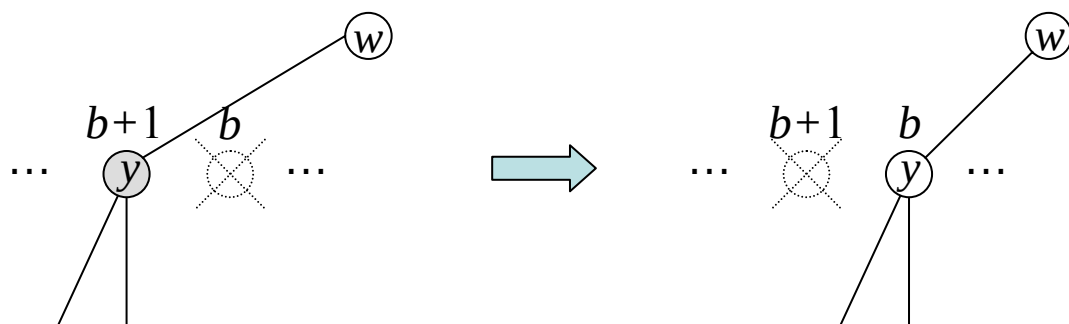
מספר תעודת זהות:

מספר מחברת:

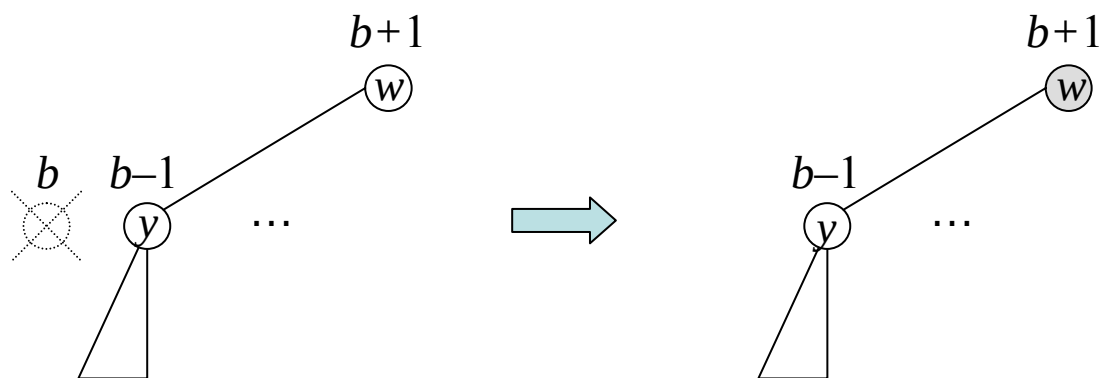
A



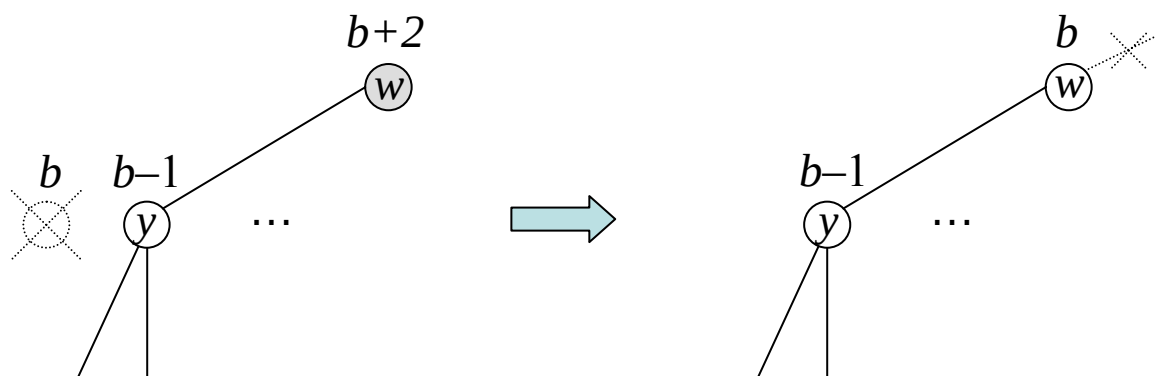
B



C



D



- א. הוכח/י כי מספר הצמתים בעץ בינומי חסר מדרגה  $k$  הוא לפחות  $F_{k+1}$  כאשר  $F_1=F_2=1$  ו- $F_k=F_{k-1}+F_{k-2}$  עבור  $k \geq 3$ . הסק/י שדרגתו של עץ בינומי חסר עם  $n$  צמתים היא  $O(\log n)$ .

- ב. מהו הזמן במקרה הגרוע של פעולת ה-decrease-key? הוכח/י את תשובתך.

מועד ג'

8 מתוך 16

מספר תעודת זהות:

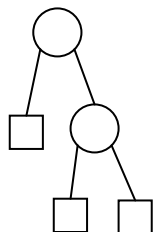
מספר מחברת:

ג. מהו הזמן amortized שלוקחת כל פעולה? שים/י לב שמימוש שאר הפעולות זהה לחלוטין למימושן בערמות פיבונצ'י? הוכח/י תשובתך. (רמז: חשוב על צומת חסר כצומת מסומן).

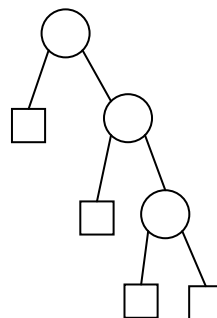


**שאלה 4 (15 נקודות)**

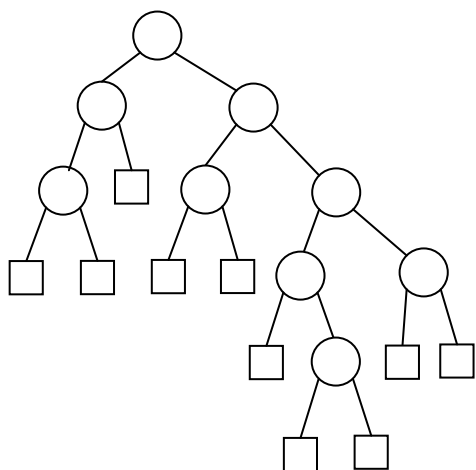
א. צבעי כל אחד מהצמתים בעצים הבאים באדום או שחור כך שהעץ יהפוך לעץ שחור-אדום חוקי או הוכח/י שאין צביעה כזו. (סמקי B עבור צומת שחור ו-R עבור צומת אדום.)



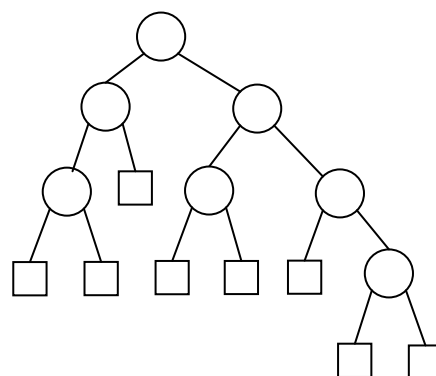
|                |        |
|----------------|--------|
| צריע / לא צריע | הוכחה: |
|                |        |



|                |        |
|----------------|--------|
| צריע / לא צריע | הוכחה: |
|                |        |



|                |        |
|----------------|--------|
| צריע / לא צריע | הוכחה: |
|                |        |



|                |        |
|----------------|--------|
| צריע / לא צריע | הוכחה: |
|                |        |

במימוש עצי אדום-שחור (Red-black trees) אנו משתמשים, בין השאר, בפעולות מהסוגים הבאים:

|   |                   |                                   |
|---|-------------------|-----------------------------------|
| R | קריאת מצביע       | לדוגמא: $left[x]$                 |
| W | השמה למצביע       | לדוגמא: $left[x] \leftarrow y$    |
| A | קריאת צבע של צומת | לדוגמא: $color[x]$                |
| B | השמת צבע          | לדוגמא: $color[y] \leftarrow RED$ |
| C | השוואת שני מפתחות | לדוגמא: $if\ k < key[x]$          |

בניתוח שהוצג בכיתה הנחנו שעלות כל הפעולות האלו, וכן כל הפעולות הבסיסיות האחרות היא קבועה. נניח עתה שלכל אחת מחמשת הפעולות האלו עלות שונה. נניח שקריאת ערכו של מצביע דורשת R יחידות זמן, השמה למצביע דורשת W יחידות זמן, קריאת צבע דורשת A יחידות זמן, וכך הלאה. נניח גם שעלות כל פעולה אחרת היא 0.

לדוגמא, ביצוע סיבוב (rotation) דורש קריאה ושינוי של מספר קבוע של מצביעים. לכן עלות הפעולה היא  $O(R+W)$ . נשים לב ש-A, B ו-C אינם מופיעים בביטוי זה שכן ביצוע סיבוב אינו דורש קריאה וכתיבה של צבעים ואינו דורש השוואת מפתחות.

לנוחיותכם מצורפים בהמשך המקרים השונים בהם נתקלים בעת פעולת insert.

ב. עלות הפעולות find, insert במקרה הגרוע ביותר, במונחים של R, W, A, B, C וכן במונחי n, מספר האיברים בעץ בעת ביצוע הפעולה הוא:

|        | Worst-case |
|--------|------------|
| find   | $O(\quad)$ |
| insert | $O(\quad)$ |

בניתוח הפעולה insert הנח/י שמקום ההכנסה נתון ולכל צומת x מצביע  $p[x]$  להורהו.

הוכחה:

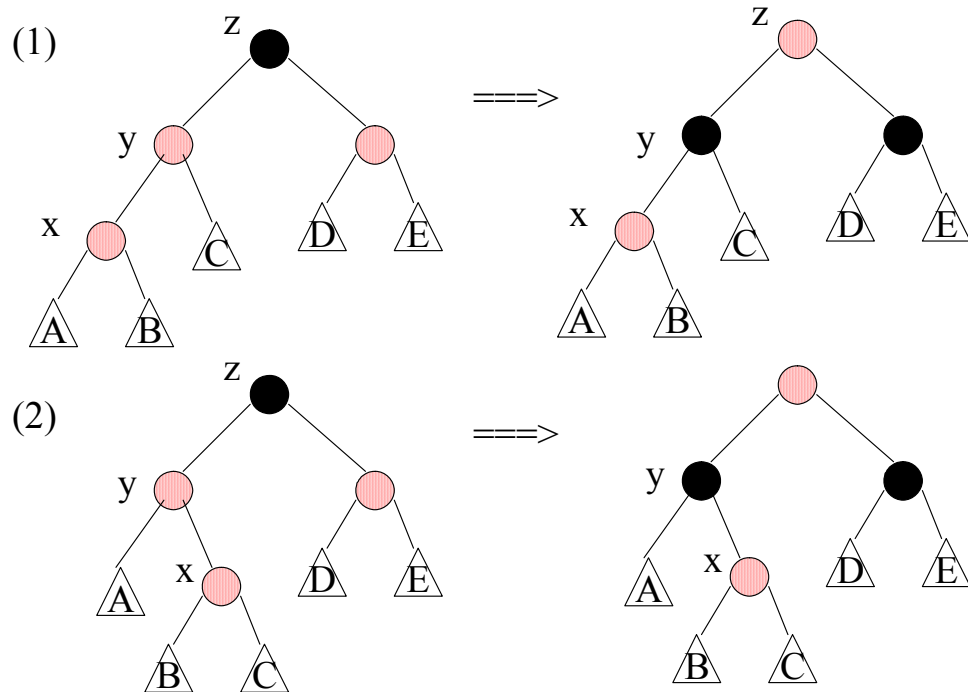
ג. עלות הפעולות find, insert במקרה ה-amortized, במונחים של  $R, W, A, B, C$  וכן במונחי  $n$ , מספר האיברים בעץ בעת ביצוע הפעולה הוא:

|        | Amortized |
|--------|-----------|
| find   | $O($ )    |
| insert | $O($ )    |

בניתוח הפעולות insert ו-delete הנח/י שמקום ההכנסה/הוצאה נתון ולכל צומת  $x$  מצביע  $p[x]$  להורוהו.

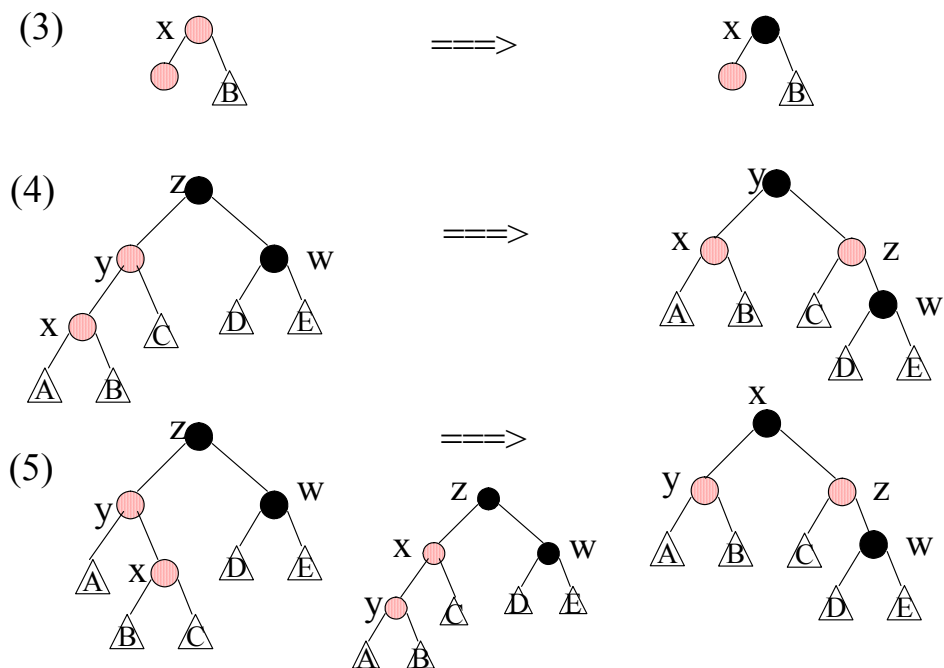
הוכחה: (ציין/י במפורש את פונקצית הפוטנציאל בה את/ה משתמש.)

## Insert -- non terminal cases



22

## Insert -- terminal cases



23

מועד ג'

13 מתוך 16

מספר תעודת זהות:

מספר מחברת:

**שאלה 5 (14 נקודות)**

א. צייר/י Generalized Suffix Tree המייצג את המחרוזות 01100 ו-101101.

ב. תאר/י אלגוריתם יעיל ככל האפשר שמקבל מחרוזת בינארית  $s$  באורך  $n$  ומוצא את המחרוזת הארוכה ביותר  $p$  כך שגם  $p$  וגם המשלים של  $p$  הן תתי מחרוזות של  $s$ . (המשלים של המחרוזת 0100 הוא לדוגמא 1011).

**שאלה 6 (14 נקודות)**

יהא  $A$  מערך בעל  $n$  מספרים. איבר  $x$  יקרא איבר רוב ב- $A$  אם הוא מופיע יותר מ- $n/2$  פעמים. לדוגמא, האיבר 3 הוא איבר רוב במערך  $[3,3,2,3,5]$ . במערך  $[1,2,1,4]$  אין איבר רוב.

א. תאר/י, במודל ההשוואות, אלגוריתם דטרמיניסטי יעיל ככל האפשר שמקבל מערך  $A$  בגודל  $n$  ומחזיר את איבר הרוב שלו, אם קיים כזה. (במידה ואין איבר רוב, על האלגוריתם לדווח על כך.) הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

בשני הסעיפים הבאים נניח שההשוואות היחידות שניתן לבצע על איברי המערך  $A$  הן השוואות מהצורה  $x=y$  שיחזירו 'כן' אם שני האיברים שווים, ו'לא' אחרת. (במידה ו- $x \neq y$  השוואה כזאת לא אומרת לנו מי גדול ממי.) להשוואות כאלה נקרא השוואות מסוג =.

ב. תאר/י, אלגוריתם דטרמיניסטי שסיבוכיותו  $O(n \log n)$ , שמשתמש רק בהשוואות מסוג  $=$ , שמקבל מערך  $A$  בגודל  $n$  ומחזיר את איבר הרוב שלו, אם קיים כזה. (במידה ואין איבר רוב, על האלגוריתם לדווח על כך.) הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

ג. תאר/י, אלגוריתם הסתברותי יעיל ככל האפשר, שמשתמש רק בהשוואות מסוג  $=$ , שמקבל מערך  $A$  בגודל  $n$  שיש בו איבר רוב ומחזיר תמיד את איבר הרוב הזה. הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

**שאלה 7 (14 נקודות)**

א. הגדר/י משפחה אוניברסלית  $H$  של פונקציות hash מקבוצה  $U$  אל  $\{0,1,\dots,m-1\}$ .

ב. תהא  $T$  טבלת hash בעלת  $m$  תאים. לטיפול בהתנגשויות משתמשים ב-chaining. ממפים קבוצת  $S$  בעלת  $n$  איברים ל- $A$  ע"י פונקציית hash  $h$  שנבחרת באקראי ממשפחה אוניברסלית של פונקציות hash. מהו  $n$  אם נתון שתוחלת מספר ההתנגשויות ב- $T$  היא  $mPP^{1/2}$ ? הוכח/י את תשובתך. (כל זוג איברים  $x \neq y \in S$  שעבורו  $h(x)=h(y)$  מגדיר התנגשות).

ג. בהמשך לב', נניח שתחת פונקציית ה-hash המסוימת שנבחרה אכן היו לכל  $m^{1/2}$  התנגשויות. כמה תאים בטבלת ה-hash יכולים להכיל לפחות  $m^{1/8}$  איברים? הוכח/י את תשובתך.

בהצלחה !!!