

# מבחן אלגוריתמים ומבנה נתונים 2

Algorithms and Data Structures 2

מבחן גמר

Final Exam

בהצלחה רבה!

Good Luck!!!

## הוראות לנבחנות

- את התשובות יש למלא במחברת המענה המצורפת. **لتשומת ליבך – טופס השאלות לא מועבר לבוחקים ולכון תשיבות שתכתבה על טופס שאלות המבחן לא תיבדקנה.**
- לפני תחילת המענה על הבדיקה – **שים לב שהפרטים האישיים האישיים המופיעים על מחברת המענה- נכונים.**
- מחברת המענה משמשת גם עבור דפי טיוטה. דפי הטיוטה לא ייסרקו ולא ייבדקו. **אנא הקפידי לרשום את התשובה שאת רוצה שיבדקו בדףים המיעודים לכך ולא בדף הטיוטה.**
- מותר לכתוב בעט או בעפפון.
- דאגי לכך שהכתב יצא ברור גם לאחר סריקה (הימנעות מכתב חלש מדי).
- משך המבחן 4 שעות.
- בimbן 4 שאלות, עלייך לענות על כולם, פירוט הניקוד של כל שאלה מופיע בכותורתה.
- חומר עזר מותר בimbן הוא חומר העזר שהודפס על ידינו ומונח על השולחנות. **אין צורך במחשבון.**
- אין לבקש/לכתוב על חומר העזר. **את חומר העזר יש להשאיר על השולחן בסוף המבחן.**
- אם הינך משתמש בתשובתך בשגרה המופיע בחומר העזר המותר בimbן, איןך צריכה להעתיקו למחברת המענה.
- אסור בהחלטת להעביר חומר עזר מאחת לרעותה. שימוש בחומר עזר משותף יגרום לפסילת הבדיקות של כל השותפות בו.
- יש לשמר על דממה מוחלטת בשעת הבדיקה. תلمידה שתדבר במהלך המבחן, מחברת הבדיקה שלה תסמן. סימון זה יילקח בחשבון בעת בדיקת המבחן.
- **דאג רב יושם לאייתור תשבות דומות מידית בין מבחנים סמוכים.**  
במקרה כזה, תיערך בדיקה מול מצלמות האבטחה, ותישקל פסילת בcheinות אלה.

**שייהיה בהצלחה :)**

# {Ultra.Code;}

**שאלה 1 (25 נקודות):** סע' א'-8 נק', סע' ב'- 9 נק', סע' ג'- 8 נק' (ג. ז - 3 נק'; ג. ו.ז - 5 נק' ))

עבור עירימת **מינימום** נתנות בהמשך השגרות  $(n \text{ MinHeap2Proc}(A, n) \rightarrow (n \text{ MinHeap1Proc}(A, n))$ .  
המקבלות מערך  $A$  המיציג את העירימה, ואת גודלה  $n$ . השגרות מבצעות שינויים באיברי העירימה, תוך שמירה על תוכנת עירימת המינימום. מדובר בערימה בינרית.

השגרות קוראות לפונקציה  $\text{Max}(A, p, r)$  מקבלת מערך  $A$  ושני אינדקסים במרקם  $p$  ו- $r$ .  
 $(1 \leq p, r \leq \text{length}(A))$ , ומחזירה את הערך המקסימלי מבין איברי  $A$  הנמצאים בתחום שבין  $p$  ל- $r$ .

## סעיף א

להלן השgraה  $(n \text{ MinHeap1Proc}(A, n))$ :

```

MinHeap1Proc( $A, n$ )
{
     $NewVal \leftarrow \text{Max}\left(A, \left\lceil \frac{n}{2} \right\rceil, n\right) + n + 1$ 
    for  $k$  from 1 to  $n$ :
        {
             $NewVal \leftarrow NewVal - 1$ 
            ChangeKey( $A, 1, NewVal$ )
        }
}

```

השgraה  $\text{MinHeap1Proc}$  קוראת לשgraה  $\text{ChangeKey}$ . כתבי, בפסאודה-קוד או בשפת תכנות כלשהי, את השgraה  $\text{ChangeKey}(A, i, x)$ , המקבלת עירימת מינימום  $A$ , אינדקס  $i$ , וערך  $x$  המקיים  $[i] < x$ , ומחליפה בערימה את  $[i]$  ב-  $x$ . השgraה  $\text{ChangeKey}$  שומרת על תוכנת עירימת המינימום.

## סעיף ב

נתחן את סיבוכיות זמן הריצה של השgraה  $(n \text{ MinHeap1Proc}(A, n))$ , ונמקן היטב את תשובהך.

## סעיף ג' מופיע בעמוד הבא

# {Ultra.Code;}

## סעיף ג

- i. השגרה  $(A, n)$  MinHeap2Proc( $A, n$ ) שלහן מחליפה את כל ערכי איברי הערימה  $A$  בערכי המשתנה המיתעדכן  $NewVal$ . שגרה זו קוראת לשגרה ChangeKey( $A, NewVal$ ) (שכתבה בסעיף א').  
השלימן את החסר בשגרה  $(A, n)$  כך שתיהה יعلاה יותר מהשגרה MinHeap1Proc( $A, n$ ).

```
MinHeap2Proc( $A, n$ )
{
     $NewVal \leftarrow \text{Max} \left( A, \left\lceil \frac{n}{2} \right\rceil, n \right)$ 
    for _____
    {
         $NewVal \leftarrow NewVal + 1$ 
        ChangeKey( $A, \underline{\hspace{2cm}}, NewVal$ )
    }
}
```

- ii. נתחן את סיבוכיות זמן הריצה של השגרה  $(A, n)$ , ונמקן היטב את תשובתך.

## המשך הבדיקה בעמודים הבאים

**שאלה 2 (20 נקודות)**

הציעי מבנה נתוניים  $S$  שבאמצעותו ניתן למשתכל את מהפעולות הבאות בסיבוכיות הזמן הנדרשת להן.  
מבנה הנתוניים המוצע יכול להיות מורכב מכמה מבני נתונים בסיסיים.

תאריך בקצרה את מבנה הנתוניים והסבירו כיצד תבוצע כל אחת מהפעולות בסיבוכיות זמן הריצה הנדרשת.

ניתן להניחס כי המפתחות שונים זה מזה.

(*a*) מצין את מספר האיברים במבנה בזמן ביצוע הפעולה.)

סיבוכיות זמן ריצה	מה מבצעת הפעולה	הפעולה
$O(\log n)$	הכנסת איבר בעל המפתח $k$ למבנה $S$	Insert( $S, k$ )
$O(\log n)$	חיפוש איבר בעל המפתח $k$ במבנה $S$	Search_Key( $S, k$ )
$O(1)$	מציאת האיבר בעל הותק הקטן ביותר במבנה $S$	Find_New ( $S$ )
$O(\log n)$	מחיקת האיבר בעל הותק הקטן ביותר במבנה $S$	Delete_New ( $S$ )
$O(1)$	הפיכת האיבר בעל הותק הגדל ביותר להיות בעל הותק הקטן ביותר במבנה	Old_Become_New( $S$ )

הערה:

תשובה מלאה היא מבנה נתונים שבאמצעותו ניתן למשתכל את כל הפעולות שלעיל בסיבוכיות הזמן הנדרשת.

תשובה חלקית היא מבנה נתונים התומך רק בביטוי חלק מהפעולות הנ"ל בסיבוכיות הזמן הנדרשת,  
ותקבל  לכל היותר ממחצית מהניקוד של השאלה.

**המשך הבדיקה בעמודים הבאים**

**שאלה 3 (30 נקודות: סע' א'-10 נק' ; סע' ב'- 20 נק' (ב.ז.- 12 נק'; ב.ז.- 8 נק'))**

סעיף א

יהי  $T_G$  עץ פורש מינימלי (MST) של גרף לא מסוון ממושקל וקשרי ( $w, E$ )  $G = \langle V, E \rangle$ .

עבור כל קשת  $T_G \in e$ , יהי  $(A_e, B_e)$  חתך של העץ  $T_G$ .

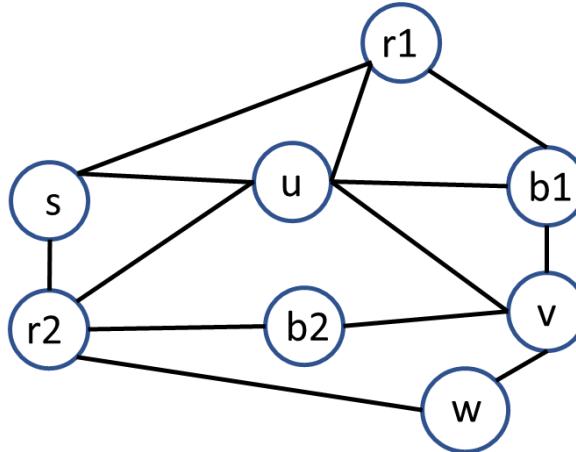
הוכיח את הטענה הבאה או הפריכי אותה על ידי דוגמא נגדית:

**לכל** עץ פורש מינימלי  $T_G$  **קיים** קשת  $T_G \in e$  **שהיא** הקשת היחידה מבין הקשות החוצות את החתך  $(A_e, B_e)$  בגרף  $G$ .

סעיף ב (סעיף זה אינו קשור לסעיף הקודם)

נתון גרף  $(V, E) = G$  לא מסוון וקשריו שבו  $5 \geq |V|$ . שניים מצמתיו הגרף צבועים באדום, ונסמנים  $\{r1, r2\}$ . שניים מצמתים אחרים בגרף צבועים בכחול, ונסמנים  $\{b1, b2\}$ . כל יתר מצמתיו הגרף צבועים בירוק. יהי הצומת  $s$  צבוע בירוק.

דוגמא לגרף המקיים אתדרישות השאלה:



נ. תארו אלגוריתם יעיל ככל הניתן, המקבל גרף  $G$  (כמפורט לעיל) וצומת  $s$  בגרף, ומוצא לכל צומת

ירוק בגרף דרך קצרה ביותר מ- $s$  אליו, העומדת בדרישות הבאות:

- הדרך עוברת לפחות באחד מהצמתים הצבועים בכחול
- הדרך אינה עוברת באף צומת הצבוע באדום

דרך בגרף יכולה להיות מסלול או שיטכן בה מעבר על קשת פעמיים.

בgraf שבדוגמה, דרך קצרה ביותר מ- $s$  ל- $w$  העומדת בדרישות:  $(s, u, b1, v, w)$

בgraf שבדוגמה, דרך קצרה ביותר מ- $s$  ל- $u$  העומדת בדרישות:  $(s, u, b1, u)$

ii. הסביר את נכונות האלגוריתם, ונתחי את סיבוכיות זמן הריצה שלו.

**שאלה 4 ( 25 נקודות : סע' א'-15 נק'; סע' ב'-10 נק' )**

**הגדירות (הקדמה לשאלה):**

- נגיד  **מחרוזת בינרית B** מעל ה-א"ב  $\{1, 0\}$ :  $B = b_1 b_2 \dots b_k$ ,  $b_i \in \{0, 1\}, 1 \leq i \leq k$
- נגיד  **סדר לקסיקוגרפי** (משמאל לימין) על מחרוזות בינריות (בדומה לסדר המיללים במילון):  
מחרוזת  $B_1$  קטנה **לקסיקוגרפית** מחרוזת  $B_2$  כאשר מתקיים אחד משני המקרים הבאים:  
1. קיימת תת-מחרוזת (רישא) באורך  $j$  (יתכן ריקה) המשותפת לשתי המחרוזות  $B_1$  ו- $B_2$  והתו  $1+j$  ב- $B_1$  הוא 0 וב- $B_2$  הוא 1.

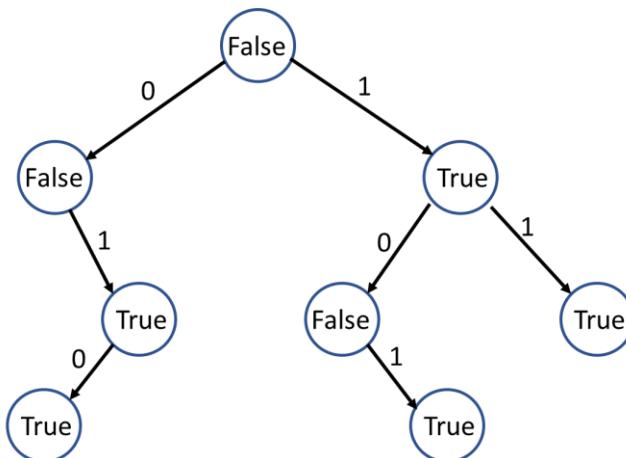
**דוגמא 1:** המחרוזת  $10011 = B_1$  קטנה **לקסיקוגרפית** מהמחרוזת  $10111 = B_2$  ( $j=2$ )

2.  $B_1$  היא תת-מחרוזת לא ריקה (רישא ממש) של  $B_2$ .

**דוגמא 2:** המחרוזת  $1001 = B_1$  קטנה **לקסיקוגרפית** מהמחרוזת  $10011 = B_2$

- נגיד  **עץ-מחרוזות-бинריות** כמבנה נתונים המאחסן קבוצת מחרוזות בינריות  $S$ . מבנה נתונים זה הוא עץ ביני שבו הקשתות מייצגות תווים, כך שקשת היוצאת מצומת אל הבן השמאלי שלו מייצגת את התו 0, וקשת אל הבן הימני מייצגת את התו 1. כל צומת מכיל ערך בוליאני (ראי  **דוגמא 3 להלן**). מסלול בעץ-מחרוזות-бинריות המתחילה בשורש העץ ומסתיים בצומת כלשהו,  **מייצג מחרוזת בינרית** בהתאם לרצף ערכי הקשתות בו.

**דוגמא 3:** מייצג את קבוצת המחרוזות:  $\{1, 01, 11, 010, 101\} = S$  בעזרת עץ-המחרוזות-הBINRIOT שלහן. מסלול באורך 2 מהשורש העובר דרך הבן השמאלי שלו, מייצג את המחרוזת 01, ומסלול באורך 3 מהשורש העובר דרך הבן השמאלי של העץ מייצג את המחרוזת 010.



# {Ultra.Code;}

השאלה:

## סעיף א

כתבו, בפסאודו-קוד או בשפת תכנות כלשהי, שגרה ( $InsertBinaryString(T, B)$ ) המתקבל מצעיר לעץ-מחרוזות-בינריות  $T$  ומחרוזת בינרית  $B$ , ומוסיפה את המחרוזת  $B$  לעץ  $T$ .  
הערה: ניתן להניח כי קיימת פונקציה  $(AllocNewNode)$  אשר מבצעת הקצת זכרון לצומת חדש בעץ ומחזירה מצביע אליו, וכן ניתן להשתמש בה.

## סעיף ב

תהיה  $\{S = \{B_1, B_2, \dots, B_m\}$  קבוצה של  $m$  מחרוזות בינריות שונות זו מזו שסכום אורכיהן הוא  $n$ , כלומר, ככלומר, מתקיים:

$$n = \sum_{i=1}^m |B_i|$$

ויהי  $T$  עץ-מחרוזות-בינריות המייצג את  $S$ . ( $|B_i|$  הוא אורך המחרוזת  $B_i$ ).

- תארו באופן מילולי אלגוריתםיעיל ככל הניתן, המקבל מצביע לעץ-מחרוזות-בינריות  $T$ , ומছזר את קבוצת המחרוזות המיוצגות בעץ כסדרה ממויינת לפי סדר לקסיקוגרפי (כמפורט לעיל).  
המשר דוגמא 3: עבור הקבוצה  $S$  המיוצגת בעץ המחרוזות הבינריות שבאיור, האלגוריתם צריך להחזיר את הסידרה הממוינת לפי סדר לקסיקוגרפי:  
 $\{01, 010, 1, 101, 11\}$ .
- סבירו את נכונות האלגוריתם ונתחי את סיבוכיות זמן הריצה שלו.

**בצלחה!**