

מבחן אלגוריתמים ומבני נתונים 2

Algorithms and Data Structures 2

מבחן גמר

Final Exam

בהצלחה רבה!

Good Luck!!!

הוראות לנבחנת

- את התשובות יש למלא במחברת המענה המצורפת. **לתשומת ליבך – טופס השאלות לא מועבר לבודקים ולכן תשובות שתכתבנה על טופס שאלות המבחן לא תיבדקנה.**
- לפני תחילת המענה על הבחינה – **שימי לב שהפרטים האישיים המופיעים על מחברת המענה-נכונים.**
- מחברת המענה משמשת גם עבור דפי טיוטה. דפי הטיוטה לא ייסרקו ולא ייבדקו. **אנא הקפידו לרשום את התשובות שאת רוצה שייבדקו בדפים המיועדים לכך ולא בדפי הטיוטה.**
- מותר לכתוב בעט או בעפרון.
- דאגי לכך שהכתב ייצא ברור גם לאחר סריקה (הימנעי מכתב חלש מדי).
- משך המבחן 4 שעות.
- במבחן 4 שאלות, עלייך לענות על כולן, פירוט הניקוד של כל שאלה מופיע בכותרתה.
- חומר עזר מותר במבחן הוא חומר העזר שהודפס על ידינו ומונח על השולחנות.
- אין צורך במחשבון.
- אין לקשקש/ לכתוב על חומר העזר.
- את חומר העזר יש להשאיר על השולחן בסוף המבחן.
- אם הינך משתמשת בתשובתך בשגרה המופיעה בחומר העזר המותר במבחן, אינך צריכה להעתיקה למחברת המענה.
- אסור בהחלט להעביר חומר עזר מאחת לרעותה. שימוש בחומר עזר משותף יגרום לפסילת הבחינות של כל השותפות בו.
- יש לשמור על דממה מוחלטת בשעת הבחינה. תלמידה שתדבר במהלך המבחן, מחברת הבחינה שלה תסומן. סימון זה יילקח בחשבון בעת בדיקת המבחן.
- **דגש רב יושם לאיתור תשובות דומות מידי בין מבחנים סמוכים.**
במקרה כזה, תיערך בדיקה מול מצלמות האבטחה, ותישקל פסילת בחינות אלה.

שיהיה בהצלחה (:

{Ultra.Code;}

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

עבור ערימת מינימום נתונות בהמשך השגרות $\text{MinHeap1Proc}(A, n)$ ו- $\text{MinHeap2Proc}(A, n)$ המקבלות מערך A המייצג את הערימה, ואת גודלה n . השגרות מבצעות שינויים באיברי הערימה, תוך שמירה על תכונת ערימת המינימום. מדובר בערימה בינרית.

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

סעיף א

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

```
MinHeap1Proc(A, n)
{
    NewVal ← Max(A, ⌊n/2⌋, n) + n + 1
    for k from 1 to n:
        {
            NewVal ← NewVal - 1
            ChangeKey(A, 1, NewVal)
        }
}
```

השגרה MinHeap1Proc קוראת לשגרה ChangeKey .

כתבי, בפסאודו-קוד או בשפת תכנות כלשהי, את השגרה $\text{ChangeKey}(A, i, x)$, המקבלת ערימת מינימום A , אינדקס i , וערך x המקיים $x > A[i]$, ומחליפה בערימה את $A[i]$ ב- x . השגרה ChangeKey שומרת על תכונת ערימת המינימום.

סעיף ב

נתחי את סיבוכיות זמן הריצה של השגרה $\text{MinHeap1Proc}(A, n)$, ונמקי היטב את תשובתך.

סעיף ג' מופיע בעמוד הבא

סעיף ג

i. השגרה $\text{MinHeap2Proc}(A, n)$ שלהלן מחליפה את כל ערכי איברי הערימה A בערכי המשתנה המתעדכן NewVal . שגרה זו קוראת לשגרה ChangeKey (שכתבת בסעיף א'). השלימי את החסר בשגרה $\text{MinHeap2Proc}(A, n)$ כך שתהיה יעילה יותר מהשגרה $\text{MinHeap1Proc}(A, n)$.

```

MinHeap2Proc(A, n)
{
    NewVal ← Max(A, ⌈ $\frac{n}{2}$ ⌉, n)
    for _____
    {
        NewVal ← NewVal + 1
        ChangeKey(A, _____, NewVal)
    }
}

```

ii. נתחי את סיבוכיות זמן הריצה של השגרה $\text{MinHeap2Proc}(A, n)$, ונמקי היטב את תשובתך.

המשך הבחינה בעמודים הבאים

שאלה 2 (20 נקודות)

הציעי מבנה נתונים S שבאמצעותו ניתן לממש כל אחת מהפעולות הבאות בסיבוכיות הזמן הנדרשת להלן. מבנה הנתונים המוצע יכול להיות מורכב מכמה מבני נתונים בסיסיים. תארי בקצרה את מבנה הנתונים והסבירי כיצד תתבצע כל אחת מהפעולות בסיבוכיות זמן הריצה הנדרשת. ניתן להניח כי המפתחות שונים זה מזה. (n מציין את מספר האיברים במבנה בזמן ביצוע הפעולה.)

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

הערה:

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

תשובה חלקית היא מבנה הנתונים התומך רק בביצוע חלק מהפעולות הנ"ל בסיבוכיות הזמן הנדרשת, ותקבל לכל היותר מחצית מהניקוד של השאלה.

המשך הבחינה בעמודים הבאים

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

סעיף א

יהי T_G עץ פורש מינימלי (MST) של גרף לא מכוון ממושקל וקשיר $G = (V, E, w)$.

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

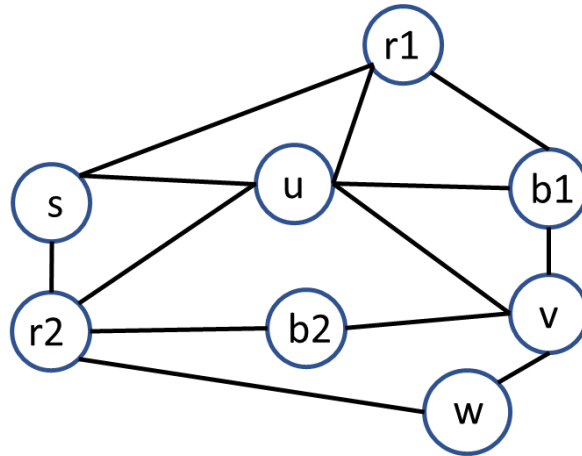
הוכיחי את הטענה הבאה או הפריכי אותה על ידי דוגמא נגדית:

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

סעיף ב (סעיף זה אינו קשור לסעיף הקודם)

נתון גרף $G = (V, E)$ לא מכוון וקשיר שבו $|V| \geq 5$. שניים מצמתי הגרף צבועים באדום, ונסמנם $\{r1, r2\}$. שני צמתים אחרים בגרף צבועים בכחול, ונסמנם $\{b1, b2\}$. כל יתר צמתי הגרף צבועים בירוק. יהי הצומת s צבוע בירוק.

דוגמא לגרף המקיים את דרישות השאלה:



i. תארי אלגוריתם יעיל ככל הניתן, המקבל גרף G (כמתואר 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 שיתכן בה מעבר על קשת פעמיים.

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

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

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

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

הגדרות (הקדמה לשאלה):

- נגדיר **מחרוזת בינרית** B מעל ה- $\{0, 1\}$:
 $B = b_1 b_2 \dots b_k$, $b_i \in \{0, 1\}, 1 \leq i \leq k$
- נגדיר **סדר לקסיקוגרפי** (משמאל לימין) על מחרוזות בינריות (בדומה לסדר המילים במילון):
מחרוזת B_1 **קטנה לקסיקוגרפית** ממחרוזת 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 ה- $j+1$ ב- B_1 הוא 0 וב- B_2 הוא 1.
דוגמא 1: המחרוזת $B_1 = 10011$ קטנה לקסיקוגרפית מהמחרוזת $B_2 = 1011$ ($j=2$).

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

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

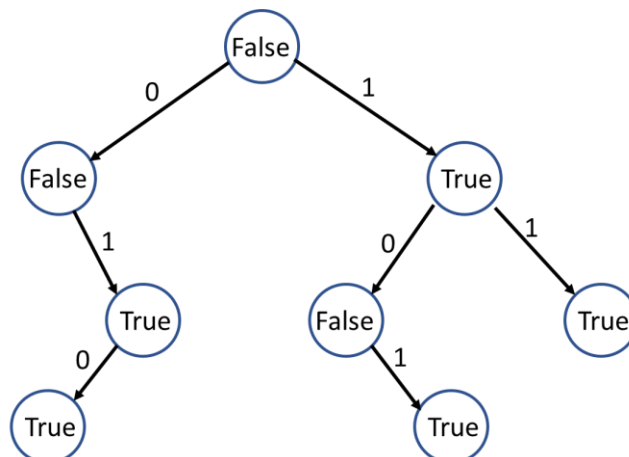
- נגדיר **עץ-מחרוזות-בינריות** כמבנה נתונים המאחסן קבוצת מחרוזות בינריות S . מבנה נתונים זה הוא עץ בינרי שבו הקשתות מייצגות תווים, כך שקשת היוצאת מצומת אל ה- 0 השמאלי שלו מייצגת את התו 0, וקשת אל ה- 1 הימני מייצגת את התו 1. כל צומת מכיל ערך בוליאני (ראי בדוגמא 3 להלן). מסלול בעץ-מחרוזות-בינריות המתחיל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

מסלול המתחיל בשורש העץ ומסתיים בצומת שערכו $True$ מייצג מחרוזת בינרית מתוך S .

מסלול המכיל רק את שורש העץ מייצג מחרוזת ריקה.

העץ יכול צמתים רק כפי הנדרש עבור קבוצת מחרוזות נתונה S .

דוגמא 3: נייצג את קבוצת המחרוזות: $S = \{1, 01, 11, 010, 101\}$ בעזרת עץ-המחרוזות-הבינריות שלהלן. מסלול באורך 2 מהשורש העובר דרך ה- 0 השמאלי שלו, מייצג את המחרוזת 01, ומסלול באורך 3 מהשורש העובר דרך ה- 0 השמאלי של העץ מייצג את המחרוזת 010.



השאלה:**סעיף א**

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

סעיף ב

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

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

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

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

בהצלחה!