מבוא למדעי המחשב
202-1-101-1

מבוא לתכנות למערכות מידע
202-1-104-1

סמסטר א' תשס"ט

<u>בוחן אמצע – 8.2.09</u>

מר איתי בר-יוסף
מר טל גרינשפון
גב' דיקלה דותן
מר שי זקוב
דר' מיכל זיו-יוקלסון
פרופ' משה זיפר
פרופ' מייק קודיש
דר' צחי רוזן

משך הבוחן – שעתיים

חומר עזר – אסור

במבחן זה 3 שאלות בניקוד שונה המסתכם ב-100 נקודות. ענו על כל השאלות.

- <u>בשאלות התכנות</u>, מספר השורות העומדות לרשותכם בדף התשובות רומז על אורך הקוד הנדרש. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא. אין צורך להעתיק את שורות הקוד הנתונות בשאלון לדף התשובות.
- **שימו לב:** החשיבות העליונה היא לנכונות הקוד. מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה שאינה יעילה או מסוגננת לא תזכה בציון המרבי.
  יש להקפיד על עימוד נכון (אינדנטציה), הוספת הערות לגבי תפקידם של משתנים נוספים עליהם הצהרתם והחזרת ערך במקום אחד בלבד בפונקציה.
- בכל השאלות, **ניתן** להשתמש בסעיפים קודמים גם במידה ולא עניתם עליהם.

בדף האחרון של המבחן ישנה תזכורת למספר פונקציות חשובות, בהן תוכלו להשתמש במהלך המבחן.
אנא רשמו את תשובותיכם **בדפי התשובות בלבד**. המחברת שקיבלתם היא מחברת טיוטה והיא לא תימסר כלל לבדיקה. בסיום הבחינה נשמור אך ורק את דף התשובות. כל שאר החומר יועבר לגריסה. תכננו את הזמן כך שתספיקו להעתיק את התשובות לדף התשובות.
הקפידו לרשום בשני דפי התשובות גם את מספר הנבחן ואת מספר החדר בו אתם נבחנים.

בהצלחה!

<u>שאלה 1 (35 נק')</u>

בשאלה זו שלושה סעיפים, העוסקים במיון של מחרוזות לפי סדר לקסיקוגרפי.
סדר לקסיקוגרפי הינו סידור של מילים לפי אותיותיהן. המילים במילון, למשל, מסודרות לפי סדר לקסיקוגרפי.

לדוגמא:
בהינתן זוג המחרוזות הבא:

```
str1 = "abca";
str2 = "abcb";
```

המחרוזת str1 קטנה לקסיקוגרפית מ str2 (מכיוון ששלושת התוים הראשונים זהים בשתי המחרוזות, ואילו התו האחרון ב str1 קודם לתו האחרון ב str2).

בהינתן זוג המחרוזות הבא:

```
str1 = "da";
str2 = "azz";
```

המחרוזת str2 קטנה לקסיקוגרפית מ str1 (מכיוון שהתו הראשון ב str2 קודם לתו הראשון ב str1).

ובהינתן זוג המחרוזות הבא:

```
str1 = "ab";
str2 = "abcd";
```

המחרוזת str1 קטנה לקסיקוגרפית מ str2 (מכיוון שהמחרוזת str1 הינה תחילית של str2).

המחרוזת הריקה קטנה לקסיקוגרפית מכל מחרוזת אחרת שאינה ריקה.

<u>שימו לב:</u> בשאלה זו **לא** ניתן להשתמש בשיטה compareTo (של המחלקה String).

<u>סעיף א' (15 נק')</u>

עליכם לממש את הפונקציה lexicoCompare, אשר מקבלת שתי מחרוזות, ומחזירה מספר שלם המציין האם המחרוזת הראשונה גדולה/קטנה/שווה למחרוזת השנייה לפי הכללים הבאים:
- במידה והראשונה גדולה לקסיקוגרפית מהשנייה, הפונקציה תחזיר מספר חיובי (למשל 1).
- במידה והראשונה קטנה לקסיקוגרפית מהשנייה, הפונקציה תחזיר מספר שלילי (למשל 1-).
- במידה והמחרוזות זהות, הפונקציה תחזיר אפס.

<u>שימו לב:</u> בסעיף זה ובכל סעיפי השאלה ניתן להניח כי המחרוזות מאותחלות (אינן null), וכי כל התוים במחרוזות הינם אותיות קטנות (a-z). כמו כן, יש לטפל תמיד גם במחרוזת הריקה.

מבנה הפונקציה:

```
public static int lexicoCompare(String str1,String str2){
    int ans;
    //                  השלימו פונקציה זו בדף התשובות, שאלה 1 א'
    return ans;
}
```

<u>סעיף ב' (8 נק')</u>

עליכם לממש את הפונקציה minIndex, אשר בהינתן מערך arr של מחרוזות ואינדקס from לאיבר כלשהו במערך, מחזירה את האינדקס של האיבר המינמלי (לפי הסדר הלקסיקוגרפי) במערך, החל מהאיבר במקום from והלאה.
למשל, קריאה לפונקציה זו, עם המערך {"abc","bcd","cde","bbb"} ועם האינדקס from=1 תחזיר את הערך 3, משום ש "bbb" קודם לקסיקוגרפית ל "bcd" ול "cde" .
ניתן לענות על סעיף זה גם במידה ולא עניתם על הסעיף הקודם.

ניתן להניח כי המערך בעל לפחות תא אחד, ו- from הינו אינדקס חוקי במערך.

מבנה הפונקציה:

```
public static int minIndex(String[] arr, int from) {
        int minInd;
        //                  השלימו פונקציה זו בדף התשובות, שאלה 1 ב'
        return minInd;
}
```

<u>סעיף ג' (12 נק')</u>

עליכם לממש את הפונקציה lexicoSort, אשר מקבלת מערך של מחרוזות, וממיינת את אבריו השונים לפי מיון בחירה. ניתן להניח שהמערך אינו null ושאורכו > 0.
זכרו כי מיון בחירה הינו זה אשר בוחר כל פעם את האיבר הקטן ביותר "הבא בתור".

<u>שימו לב:</u> אסור לכם ליצור מערך עזר נוסף או להוסיף פונקציות עזר.

```
public static void lexicoSort(String[] arr){

        //                  השלימו פונקציה זו בדף התשובות, שאלה 1 ג'

}
```

הינכם רשאים להשתמש בפונקציה swap אשר מחליפה את ערכי המערך בתאים הנתונים:

```
public static void swap(String[] arr, int i, int j) {
        String tmp = arr[i];
        arr[i] = arr[j];
        arr[j] = tmp;
}
```

<u>הקדמה לשאלה 2</u>

אם הנכם בקיאים בהקשר של עבודת הבית מס' 3 דלגו על הקדמה זו.

- פסוקית הינה disjunction ("או") של ליטרלים מהצורה:

$$l_1 \vee l_2 \vee \cdots \vee l_n \quad (n \geq 0)$$

  זכרו גם כי כאשר n=0 הפסוקית שקולה ל-false.
- נוסחא בתחשיב הפסוקים הינה בצורת CNF אם צורתה היא conjunction ("וגם") של פסוקיות:

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m \quad (m \geq 0)$$

  זכרו גם כי כאשר m=0 הנוסחא שקולה ל-true.
- השמה היא התאמה של ערכי אמת/שקר למשתנים הבוליאנים.
- השמה מספקת נוסחא אם ערך הנוסחא הוא "אמת" כאשר מציבים את ערכי ההשמה למשתנים.
- בשאלה זו כמו בעבודת הבית מס' 3, אנו מייצגים פסוקית של תחשיב הפסוקים כמערך של מספרים שלמים (int[]), כאשר ערך האיבר מציין האם הליטרל חיובי או שלילי.
  למשל, הפסוקית $x_1 \vee \overline{x_2} \vee \overline{x_3}$ תיוצג ע"י המערך {1, -2, -3}.
  נוסחת CNF מיוצגת כמערך של פסוקיות (מערך דו-מימדי של מספרים שלמים – [][]int).

שאלה 2 (30 נק')

פסוקית CNF תקרא חיובית אם ההשמה הנותנת את הערך true ("אמת") לכל המשתנים מספקת את הפסוקית.
נוסחת CNF תקרא חיובית אם ההשמה הנותנת את הערך true ("אמת") לכל המשתנים מספקת את הנוסחא.

סעיף א' (15 נק')

השלימו הפונקציה isPositiveClause אשר מקבלת פסוקית c, ומחזירה את הערך true אם ורק אם c הינה פסוקית חיובית.

ניתן להניח כי c היא פסוקית חוקית, כלומר, c איננה null ואינה מכילה את הערך 0. שימו לב כי c עשויה להיות ריקה (ללא משתנים).

```
public static boolean isPositiveClause(int[] c){
        boolean ans;
        //                  השלימו פונקציה זו בדף התשובות, שאלה 2 א'
        return ans;
}
```

לדוגמא, עבור הפסוקיות הבאות:

```
int[] c1 = { 1, -2 };
int[] c2 = { -1, -2, -3 };
```

c1 היא פסוקית חיובית כיוון שההשמה $\mu = \{x_1 \mapsto true, x_2 \mapsto true\}$ מספקת את הפסוקית ולכן isPositiveClause(c1) תחזיר true. לעומת זאת c2 איננה פסוקית חיובית כיוון שההשמה $\mu = \{x_1 \mapsto true, x_2 \mapsto true, x_3 \mapsto true\}$ איננה מספקת את הפסוקית, ולכן isPositiveClause(c2) תחזיר false.

סעיף ב' (15 נק')

השלימו את הפונקציה isPositiveCnf אשר מקבלת נוסחת CNF חוקית phi, ומחזירה true אם ורק אם phi הינה נוסחת CNF חיובית.

ניתן להניח כי הנוסחא המתקבלת הינה נוסחת CNF חוקית, כלומר, phi איננה null וכל פסוקית ב- phi חוקית (כמו בסעיף א'). שימו לב כי phi עשויה להיות ריקה (ללא פסוקיות).

```
public static boolean isPositiveCnf(int[][] phi){
        boolean ans;
        //                  השלימו פונקציה זו בדף התשובות, שאלה 2 ב'
        return ans;
}
```

לדוגמא, עבור הנוסחאות הבאות:

```
int[][] phi1 = { { 1, -2 }, { 2 } };
int[][] phi2 = { { 1, 2 }, { -1 } };
```

phi1 היא נוסחת CNF חיובית כיוון שההשמה $\mu = \{x_1 \mapsto true, x_2 \mapsto true\}$ מספקת את הנוסחא ולכן isPositiveCnf(phi1) תחזיר true. לעומת זאת, phi2 איננה נוסחת CNF חיובית כיוון שההשמה $\mu = \{x_1 \mapsto true, x_2 \mapsto true\}$ לא מספקת את הנוסחא ולכן isPositiveCnf(phi2) תחזיר false.

שאלה 3 (35 נק')

ניתן לייצג תמונה באמצעות מפת ביטים (bitmap). המפה היא למעשה מערך דו מימדי של תאים – פיקסלים (pixel קיצור של picture cell). כל פיקסל נושא צבע. לשם הפשטות החלטנו לייצג את הצבעים השונים באמצעות מספרים שלמים (int).

סעיף א' (10 נק'):

השלימו את הפונקציה inImage (int [][] image, int i, int j), המקבלת תמונה המיוצגת כמערך דו-מימדי וכן קואורדינאטות, ומחזירה ערך אמת רק במידה והקואורדינאטות הינן בתוך התמונה. קואורדינאטת ה i מתאימה למימד הראשון במערך וקואורדינאטת ה j למימד השני.
בסעיף זה ובכל סעיפי השאלה ניתן להניח כי image אינו null וכי הוא מייצג תמונה בצורה מלבנית כלשהי (במימדים הגדולים מ 1^a1).

```
public static boolean inImage (int [][] image, int i, int j) {
        boolean ans;
        //                  השלימו פונקציה זו בדף התשובות, שאלה 3 א'
        return ans;
}
```

לדוגמא, המערך הדו מימדי הבא: `int[][] image = { {1,2}, {2,2}, {1,4} };`

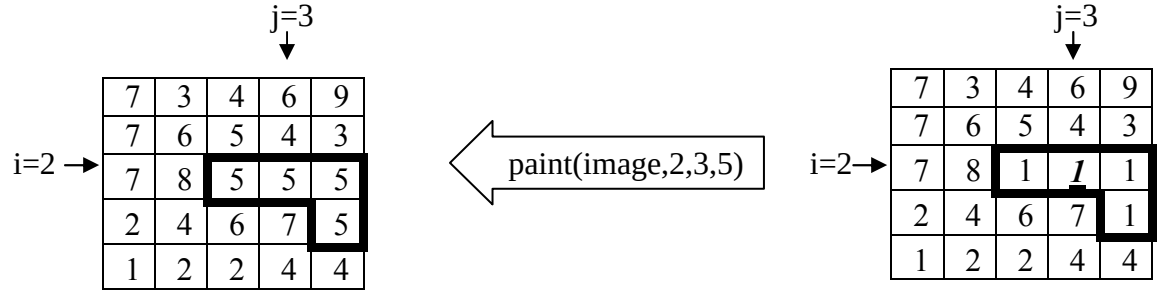
מייצג את התמונה הבאה:

1	2
2	2
1	4

בתמונה זו, הקואורדינאטה i=1, j=2 נמצאת מחוץ לתמונה, והקואורדינאטה i=2, j=1 נמצאת בתוך התמונה וערכה 4.

סעיפים ב'-ד' (25 נק')

סעיפים אלו עוסקים בצביעת שטח בתוך תמונה. צביעת השטח מתפשטת החוצה החל מהנקודה שנבחרה, לכל ארבעת התאים מסביב שיש להם את אותו הצבע של הנקודה (בתוך גבולות התמונה כמובן). הצבע אינו מתפשט באלכסון.
עליכם לממש את הפונקציה paint, המקבלת תמונה image המיוצגת כמערך דו-ממדי, קואורדינאטות i ו- j וצבע color, ומשנה את צבעו של האזור הצבוע בצבע של התמונה בקואורדינטות i ו- j, לצבע color.
לדוגמא, לאחר הפעלת paint עם i=2, j=3, color=5 על התמונה מימין, נקבל את התמונה משמאל:



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

7	3	4	6	9
7	6	5	4	3
7	8	1	***1***	1
2	4	6	7	1
1	2	2	4	4

השלימו את הפונקציה **<u>הרקורסיבית</u>** paintRec ואת הפונקציה paint הקוראת לה.

```
public static void paint(int[][] image, int i, int j, int color){
        if (inImage(image,i,j))
                paintRec(image,i,j, /*  השלימו בדף התשובות, שאלה 3 ב' */);

}

public static void paintRec(/*  השלימו בדף התשובות, שאלה 3 ג' */){
        //                  השלימו פונקציה זו בדף התשובות, שאלה 3 ד'
}
```

<u>מומלץ לבדוק את הקוד שכתבתם ע"י הרצה ידנית של דוגמא כלשהי.</u>

בהצלחה !

<u>תזכורת:</u>

- `int` **`length`**`()`

שיטה במחלקה String המחזירה את אורך המחרוזת

- `int` **`indexOf`**`(char c)`

שיטה במחלקה String, המחזירה את האינדקס הראשון במחרוזת, בו מופיע התו c, או 1- במידה והתו כלל לא מופיע במחרוזת.

- `char` **`charAt`**`(int index)`

שיטה במחלקה String, המחזירה את התו במיקום index במחרוזת

- `String` **`substring`**`(int beginIndex)`

שיטה במחלקה String המחזירה מחרוזת חדשה, שהיא תת-מחרוזת של עצם המפתח החל מהתו beginIndex ועד סוף המחרוזת.

- `String` **`substring`**`(int beginIndex, int endIndex)`

שיטה במחלקה String המחזירה מחרוזת חדשה, שהיא תת-מחרוזת של עצם המפתח בין התווים beginIndex ו-endIndex, (כולל התו ה-beginIndex ולא כולל התו ה-endIndex)

- `double` **`Math.pow`**`(double a, double b)`

פונקציה המחזירה את הערך של a בחזקת b

- `double` **`Math.sqrt`**`(double a)`

פונקציה המחזירה את השורש הריבועי החיובי של a

- `double` **`Math.abs`**`(double a)`

פונקציה המחזירה את ערכו המוחלט של a

- `int` **`Math.abs`**`(int a)`

פונקציה המחזירה את ערכו המוחלט של a

- `int` **`Math.min`**`(int a, int b)`

פונקציה המחזירה את הערך המינימלי מבין a ו- b

- `double` **`Math.min`**`(double a, double b)`

פונקציה המחזירה את הערך המינימלי מבין a ו- b

- `int` **`Math.max`**`(int a, int b)`

פונקציה המחזירה את הערך המקסימלי מבין a ו- b

- `double` **`Math.max`**`(double a, double b)`

פונקציה המחזירה את הערך המקסימלי מבין a ו- b

- `double` **`Math.random`**`()`

פונקציה המחזירה ערך אקראי בטווח 0 (כולל) עד 1 (לא כולל)