

מבוא למדעי המחשב 202-1-101-1

סמסטר א' תשס"ג
בוהן מחצית הסמסטר

מר בועז בן-משה
דר' יצחק רוזן
פרופ' מיכאל קודיש
דר' חן קיסר

משך הבוחן שעתיים וחצי
חומר עזר - אסור
אין להשתמש במחשבון.

במבחן זה 6 שאלות, בניקוד המסתכם ב- 100 נקודות. ענו על כל השאלות.

אנא רשמו את תשובותיכם בדף 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. הקפידו לרשום בדף התשובות גם את מספר הנבחן ומספר החדר שבו אתם נבחנים.

בשאלות התכנות, 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.**

בהצלחה !

שאלה 1 (10 נקודות סה"כ, 1-2 לכל סעיף)

סעיף א' נסחו מחדש את השורה המסומנת ב-(a) כך שלא תשתמש באופראטור &&.

```
public static boolean aleph(int[] a, int j){
    return j < a.length && a[j] > 0; //(a)
}
```

סעיף ב' נסחו מחדש את השורה המסומנת ב-(b) כך שלא תשתמש באופראטור ||.

```
public static boolean bet(int[] a, int j){
    return j >= a.length || a[j] <= 0; //(b)
}
```

הסעיפים הבאים מתייחסים לשיטה `boolean implies(boolean p, boolean q)` אשר מממשת את אופראטור הגרירה הלוגית $(p \rightarrow q)$. בכל אחד מהסעיפים יש לציין מה תדפיס תוכנית שמכילה את שורות הקוד המצוינות. במידה ותהיה שגיאת הרצה יש לציין "error".

```
public static boolean implies(boolean p, boolean q){
    return ((!p) || q);
}
```

סעיף ג'

```
System.out.println(implies(false,true));
```

סעיף ד'

```
System.out.println(implies(true,false));
```

סעיף ה'

```
int[] a = {1,1,1,1,1}; int j = 10;
System.out.println(!(j<a.length) || a[j]==1 );
```

סעיף ו'

```
int[] a = {1,1,1,1,1}; int j = 10;
System.out.println(implies(j<a.length, a[j]==1));
```

```

public class WhatDoIPrint{
    static int i1 = 0;
    static int j = 1;
    public static void main(String[] a) {
        int i1 = 123;
        int[] i2 = {i1};
        int[][] i3 = {i2};
        i1 = 321;
        System.out.println(i1+" "+i2[0]+" "+i3[0][0]);           //a
        i2[0] = i1;
        System.out.println(i1+" "+i2[0]+" "+i3[0][0]);           //b
        System.out.println(mystery1(i1)+" "+i1+" "+j);           //c
        System.out.println(i1+" "+j+" "+ mystery1(i1));           //d
        System.out.println(i2[0]+" "+j+" "+ mystery1(i2[0]));    //e
        System.out.println(i2[0]+" "+j+" "+mystery2(i2[0]));    //f
        System.out.println(i2[0]+" "+(mystery3(i2)==i2)+
            " "+i2[0]);                                           //g
    }
    static int mystery1(int i1) {
        int i4 = i1+j;
        i1 = i1 + 1;
        j  = j + 1;
        return i4;
    }
    static int mystery2(int j) {
        int i4 = i1+j;
        i1 = i1 + 1;
        j  = j + 1;
        return i4;
    }
    static int[] mystery3(int[] j) {
        int[] i4 = j;
        i4[0] = j[0]+5;
        return i4;
    }
}

```

שאלה 3 (20 נקודות)

נתונה התוכנית Search המכילה את הפונקציה recSearch, שאמורה לבצע חיפוש בינארי במערך של מספרים שלמים ממוינים וכן פונקצית main הבודקת אותה. יתכן ויש שגיאות בתוכנית.

```
public class Search{
    public static void main (String[] args)  {
        int len = 10;
        int[] arr = new int[len];
        for(int i = 0 ; i < len ; i=i+1){
            arr[i] = i;
        }
        int x = (int) (Math.random()*11); // that means: 0<=x<=10
        System.out.println(x + ", "+
            recSearch(x,arr,0,arr.length)); // (1)
    }

    public static int recSearch(int e, int[] a, int from, int till){
        if (from <= till) { // (2)
            int i = (from + till)/2;
            if (e < a[i]) {
                return recSearch(e,a,from,i-1); // (3)
            }
            else if (e > a[i]) {
                return recSearch(e,a,i+1,till);
            }
            else return i;
        }
        return -1;
    }
}
```

להלן מספר משפטים. סמנו בדף התשובות את המשפטים הנכונים. מספר המשפטים הנכונים עשוי להיות שונה מאחד, יתכן אפילו שכולם נכונים.

- א. התוכנית עוברת קומפילציה ומדפיסה תוצאה בכל הרצה. אם $x=10$ יודפס -1.
- ב. התוכנית עוברת קומפילציה ובמקרה ש- $x=10$ נתקלת בשגיאת הרצה (run time error).
- ג. התוכנית עוברת קומפילציה ובכל המקרים שבהם אין שגיאת הרצה מדפיסה שני מספרים זהים. לדוגמה (3, 3).
- ד. אם נחליף (בתוכנית המקורית) בשורה המסומנת (2) את הסימן \leq בסימן $<$ אזי בכל הרצה שבו $x < 10$ התוכנית תדפיס שורה ובה שני מספרים זהים.
- ה. אם נחליף (בתוכנית המקורית) בשורה המסומנת (1) את `arr.length` ב- `arr.length-1` אזי התוכנית תתבצע ללא שגיאת הרצה עבור כל ערך של x .
- ו. אם נחליף (בתוכנית המקורית) בשורה המסומנת (3) את $i-1$ ב- i אזי בכל הרצה שבו $x < 10$ התוכנית תדפיס שורה ובה שני מספרים זהים.
- ז. אף אחד מהמשפטים א-ו אינו נכון.

שאלה 4 (16 נקודות)

סעיף א' (6 נקודות)

השלימו את הגדרת השיטה הבאה אשר מקבלת מערך a של מספרים שלמים ומחזירה, אם אפשר, אינדקס i כך ש- $a[i] > a[i+1]$. אם אי-אפשר, אז על השיטה להחזיר -1.

```
public static int findIndex(int[] a){
    int answer = -1;
    //
    //                      השלימו בדף התשובות
    //
    return answer;
}
```

לדוגמה: אם נגדיר

```
int[] arr = {1,2,4,3,0};
```

אזי על `findIndex(arr)` להחזיר את אחד הערכים 2 או 3 (שתי האפשרויות תקינות).

סעיף ב' (10 נקודות)

היזכרו בשיטה הבאה, אשר מקבלת מערך a ושני אינדקסים i ו- j של תאים עליו, ומחליפה את הערכים בשני התאים.

```
static void swap(int [] a, int i, int j){
    int tmp=a[i]; a[i]=a[j]; a[j]=tmp;
    return;
}
```

לדוגמה, אחרי ביצוע `swap(arr,0,4)` המערך מהדוגמה שבסעיף הקודם ייראה כך: $\{0,2,4,3,1\}$.

הבחנה: אם נזהה אינדקס i במערך a כך ש- $a[i] > a[i+1]$ ונחליף את הערכים בתאים i ו- $i+1$ אזי המערך שיתקבל יהיה "יותר ממויין" מאשר קודם לכן. בפרט, אם לא קיים אינדקס $0 \leq i < arr.length-1$ כך ש- $a[i] > a[i+1]$ אזי המערך הנו ממויין.

השלימו את השיטה הבאה, אשר מממשת אלגוריתם של מיון על בסיס ההבחנה. יש להשתמש בשיטות `findIndex` ו-`swap` (מותר להשתמש בשיטה `findIndex` גם אם לא עניתם על סעיף א').

```
public static void Sort(int[] a){
    //
    //                      השלימו בדף התשובות
    //
}
```

שאלה 5 (20 נקודות)

הפונקציה `String[] count(int n)` מקבלת מספר שלם n גדול או שווה ל-0 (מותר להניח קלט תקין) ומחזירה מערך של כל 2^n המחזורות באורך n המורכבות מאפסים ואחדים. סדר המחזורות הוא סדר המניה מאפס ל- $(2^n - 1)$ על פי בסיס 2. למשל:

`count(0)` יחזיר `{}"`
`count(1)` יחזיר `{"0","1"}`
`count(2)` יחזיר `{"00","01","10","11"}`
`count(3)` יחזיר `{"000","001","010","011","100","101","110","111"}`

השלימו את הגדרת השיטה. הפתרון חייב להיות רקורסיבי.

```
public static String[] count(int n) {
    String[] answer;
    if (n==0) {
        answer = new String[1];
        answer[0] = "";
    }
    else {
        //
        //          השלימו בדף התשובות
        //
    }
    return answer;
}
```

שאלה 6 (20 נקודות, עשר נקודות לכל סעיף)

המחלקה IntegerSet מייצגת קבוצה של מספרים שלמים ותומכת בפעולות על קבוצות. השלימו את שיטות ובוני המחלקה.

אין להשתמש בשיטות ובונים של המחלקה שאינם מצוינים במפורש.
מותר להשתמש גם בשיטה שלא השלמתם.

```
public class IntegerSet {

    private Integer[] internalArray;
    private int size;

    public IntegerSet() {
        // יוצר קבוצה ריקה.
        // אין צורך להשלים.
    }

    public boolean isMember(Integer element) {
        // מחזירה אמת אם ורק אם element הוא איבר בקבוצה.
        // אין צורך להשלים.
    }

    public void add(Integer element) {
        // מוסיפה איבר לקבוצה. במידת הצורך מגדילה
        // את המערך.
        // אין צורך להשלים.
    }

    public size() {
        return size;
    }

    public IntegerSet intersect(IntegerSet otherSet) {
        IntegerSet answer = new IntegerSet();
        // א. מחזירה את קבוצת החיתוך עם otherSet.
        // מבלי לשנות את עצם המפתח (העצם הפועל)
        // השלימו בדף התשובות
        //
        return answer;
    }

    public boolean equals(IntegerSet otherSet) {
        boolean answer = false;
        // ב. מחזירה true אם ורק אם שתי הקבוצות מכילות
        // אותם אברים. השלימו בדף התשובות
        //
        return answer;
    }
}
```

מבוא למדעי המחשב
בוחרן אמצע 6/12/2002

תיקונים והבהרות

- א. בשאלה 1, סעיפים א' ו- ב', עליכם להחליף את השורות המסומנות (a) ו- (b) בקטעי הקוד **מבלי להשתמש** באופרטורים && ו/או ||. על תשובתיכם לפעול באותו אופן כמו הקוד הנתון.
- ב. בשאלות 2 ו- 3, סדר הסעיפים כפי שהוא מופיע בדף התשובות עלול להיות קצת מבלבל. וודאו שאתם עונים על כל סעיף במקום המיועד לו.
- ג. בשאלה 5 שימו לב שהפונקציה צריכה להחזיר מערך **מסודר לפי סדר מניה עולה**, כלומר 0,1,2,3... בייצוג בינרי (התבוננו בדוגמאות).
- ד. הניחו בשאלה 6 (בשני הסעיפים) שהקלט לפונקציות (מסוג IntegerSet) אינו null.

דף עזר

- א. להזכירכם, הפונקציה Math.random() מחזירה מספר מטיפוס double בין 0 ל- 1 כולל 0 ולא כולל 1.
- ב. החישובים בהפעלת האופרטור + (חיבור, שרשור וכו') מתבצעים משמאל לימין. לדוגמא אם נתונות שיטות g ו-f שמחזירות מחרוזת, אזי בהרצת הפקודה:
 $f() + g();$
סדר הפעולות יהיה:
- קריאה ל f()
 - קריאה ל g()
 - הפעלת האופרטור + על התוצאות

מבוא למדעי המחשב
בוהן אמצע 6/12/2002

במהלך הבחינה נענו השאלות הבאות:

- בשאלה 6 הייצוג הפנימי של קבוצה מניח שהערכים מופיעים פעם אחת.
- בשאלה 6, size – מספר האיברים בקבוצה. השיטה size() מחזירה int.
- בשאלה 4 מותר להניח שהמערך שונה מ-null.