

מבוא למדעי המחשב
202-1-101-1

סמסטר ב', תשס"ח, 2007/8

מבחן מועד א

דר' צחי רוזן
מר אילן כדר

משך הבחינה: שלוש שעות (לא תהיה הארכת זמן)
חומר עזר: אסור

בבחינה 5 שאלות. ענו על כולן: הניקוד מסתכם ב-100 נקודות.

בשאלות התכנות החשיבות העליונה בד"כ היא על נכונות הקוד, מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגגנת תזכה בציון גבוה יותר מאשר תשובה מסורבלת או ארוכה מדי.

בכל השאלות אתם רשאים להשתמש בסעיפים קודמים, אם כי לא בהכרח תזדקקו לכך. אתם רשאים לעשות זאת אפילו אם לא עניתם על הסעיפים האמורים.

לבחינה מצורף בסופה דף תשובות. רשמו את תשובותיכם אך ורק בדף התשובות במקומות המיועדים. המחברת שקיבלתם היא מחברת טיוטה בלבד, והיא וטופס הבחינה עצמו לא יימסרו כלל לבדיקה.

שימו לב: מספר השורות הריקות בדף התשובות בהחלט מרמז על אורך התשובה.

הקפידו על כתב יד ברור.

הקפידו לרשום את מספר הנבחן על דף התשובות.

בהצלחה!

שאלה 1 (20 נקודות)

סעיף א (10 נקודות):

וקטור באורך n נקרא מושלם אם הוא מכיל את כל המספרים $1 \dots n$.

למשל כל אחד מהוקטורים הבאים הוא מושלם.

```
{1, 2, 3}
{3, 1, 2}
{2, 3, 1}
```

ללא שימוש במערכי עזר השלימו בדף התשובות את השיטה

```
boolean isPerfect (int [] vec)
```

שמקבלת וקטור vec באורך n ובודקת האם הוא מושלם.

שימו לב: וקטור באורך אפס הוא וקטור מושלם.

הניחו ש- $vec \neq null$.

סעיף ב (10 נקודות):

השלימו בדף התשובות את השיטה

```
public static boolean isInclude(int vec[], int u)
```

השיטה מקבלת וקטור vec ממוין בסדר לא יורד ומספר u שלם ולא שלילי וקובעת האם הווקטור מכיל את סדרת המספרים $1^2, 2^2, \dots, u^2$.

שאלה 2 (20 נקודות)

סעיף א (10 נקודות):

וקטור vec באורך n נקרא מאוזן סביב מקום $0 < p < n$ אם $\sum_{i=0}^{p-1} vec[i] = \sum_{i=p}^{n-1} vec[i]$. למשל הווקטור $\{5, 6, 1, 2, 8\}$ מאוזן סביב מקום 2.

השלימו את השיטה הרקורסיבית הבאה בדף התשובות.

```
int where(int[] vec, int l, int p, int r)
```

השיטה מקבלת ארבעה פרמטרים:

- vec - ווקטור.
- l - האינדקס של המקום השמאלי ביותר ב- vec ,
- p - אינדקס של מקום ב- vec שסביבו vec עשוי להיות מאוזן,
- r - האינדקס של המקום הימני ביותר ב- vec .

בהנחה שהקריאה הראשונה היא $where(vec, 0, 1, vec.length-1)$, השיטה מחזירה מקום שסביבו vec מאוזן. אם אין מקום כזה השיטה מחזירה -1.

הניחו שהפרמטרים תקינים.

הניחו שעומדת לרשותכם שיטה מהצורה

```
int sum (int[] vec, int l, int r)
```

שמחזירה את $\sum_{i=l}^r vec[i]$.

סעיף ב (10 נקודות):

הגדרה:

- וקטור vec נקרא k -מאוזן אם הוא מאוזן סביב איזשהו מקום $0 \leq p < n$ (עפ"י ההגדרה שבסעיף א) ואם תת-הווקטורים $vec[0 .. p-1]$ ו- $vec[p .. n-1]$ $(k-1)$ -מאוזנים.
- כל ווקטור הוא 0-מאוזן.

למשל הווקטור $\{1, 2, 2, 1, 3, 2, 1\}$ הוא 2-מאוזן שכן:

- הוא מאוזן סביב 4
- והווקטורים $\{1, 2, 2, 1\}$ ו- $\{3, 2, 1\}$ הם 1-מאוזנים.

השלימו בדף התשובות את השיטה

`boolean isBalanced(int[] vec, int l, int r, int k)`

שמקבלת ווקטור `vec` באורך `n` ושני אינדקסים $0 \leq l \leq r < n$ ומספר לא שלילי `k` ובודקת האם `vec[l..r]` הוא `k`-מאוזן.

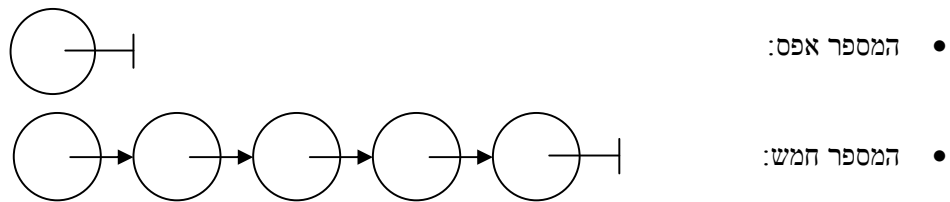
הניחו שהפרמטרים תקינים.

שאלה 3 (20 נקודות)

בעולם בו יש רק עצמים (מסוג אחד), שהתכונה היחידה שלהם שהם יכולים להכיל עצמים אחרים, ניתן לייצג מספרים טבעיים באופן הבא:

- אפס ע"י העצם שלא מכיל אף עצם אחר.
- מספר $n > 0$ ע"י העצם שמכיל את העצם שמייצג את המספר $n-1$.

בצורה גרפית:



להלן מחלקה בשם Nat שהעצמים שלה מתנהגים באופן המתואר למעלה.

```
public class Nat {  
  
    private final Nat pre;  
    public Nat (Nat pre) { this.pre = pre; }  
    public Nat () { this(null); }  
    public boolean isZero () { return pre == null; }  
    public Nat succ () { return new Nat(this); }  
  
    ...  
}
```

השלימו בדף התשובות את השיטות הבאות של המחלקה Nat: שימו לב: פתרונות שימירו את האובייקטים למספרים שלמים לא יזכו בנקודות

סעיף א (10 נקודות):

```
public boolean equals (Nat other)
```

השיטה מחזירה true אם העצם הנושא מייצג את אותו מספר ש- other מייצג.

סעיף ב (10 נקודות):

```
public Nat plus (Nat other)
```

השיטה מחזירה עצם שמייצג את הסכום של המספרים שעצם הנושא ו- other מייצגים. למשל אם עצם הנושא מייצג את 5 ו- other מייצג את 2, על השיטה להחזיר עצם שמייצג את 7.

שאלה 4 (25 נקודות)

המחלקה Node מייצגת קודקוד של עץ בינארי והיא מוגדרת כלהלן:

```
public class Node {
    public Object data;
    public Node parent, left, right;
}
```

- data – הנתונים.
- left – הבן שמאלי. null אם אין בן שמאלי.
- right – הבן ימני. null אם אין בן ימני.
- parent – ההורה. null אם שורש.

המחלקה BTree מייצגת עץ חיפוש בינארי והיא מוגדרת כלהלן:

```
public class BTree {
    protected Node root;
}
```

- root – שורש העץ. null אם העץ ריק.

טעיף א (10 נקודות):

הממשק Iter מייצג אובייקט, דוגמת מערך ממורן, רשימה מקושרת ממוינת או עץ חיפוש בינארי, שניתן לעבור על איבריו בזה אחר זה בסדר לא יורד והוא מוגדר כלהלן:

```
public interface Iter {
    void reset();
    boolean hasNext();
    Object next();
}
```

- reset – מתחיל מעבר חדש על איברי האובייקט.
- hasNext – מחזיר true אם יש עוד איברים של האובייקט שעדיין לא עברנו עליהם מאז ה-reset האחרון. **שימו לב:** אין צורך לטפל במקרה שלא נעשה reset בכלל.
- next – מחזיר את האיבר הבא.

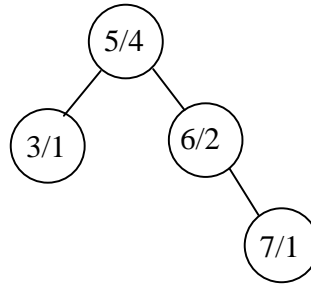
ללא הוספת שדות (מעבר לשדה current שלהלן) השלימו את המחלקה הבאה בדף התשובות:

```
public class IterBTree extends BTree implements Iter {
    private Node current;
}
```

הניחו שבין תחילה לבין סיום מעבר כלשהו לא נעשה שינוי באובייקט.

סעיף ב (15 נקודות):

k-עץ-חיפוש-בינארי הוא עץ חיפוש בינארי שבכל קודקוד x שלו בנוסף לנתונים נשמר גם מספר הקודקודים בתת-העץ המושרש ב-x. למשל, בדוגמה הבאה בכל צומת שני ערכים a/b, כאשר a הוא מפתח ו-b הוא גודל תת-העץ.



המחלקה KBSTree שלהלן מייצגת k-עץ-חיפוש-בינארי של ערכים שלמים **שונים** בגודלם:

```
public class KBSTree {  
  
    private class Node {  
        Integer data;  
        int size;  
        Node left, right;  
    }  
  
    private Node root;  
}
```

הוסיפו בדף התשובות למחלקה KBSTree את השיטה הבאה:

```
public Integer kmax(int k)
```

השיטה מקבלת מספר שלם חיובי k ומחזירה את הערך ה-k בגודלו של עץ-חיפוש הבינארי. למשל, אם עצם הנושא הוא העץ שלמעלה ו-k = 1, על השיטה להחזיר את 3. אם k = 3 על השיטה להחזיר את 6.

הניחו שהפרמטר k תקין ושהוא בתחום 1 .. n, כאשר n הוא מספר הערכים בעץ החיפוש.

שימו לב: רק פתרון שישתמש בשדה size יזכה במלוא הנקודות אתם רשאים להוסיף פונקציה פרטית (אחת או יותר) כרצונכם.

שאלה 5 (15 נקודות)

עיינו בקוד הבא ועבור כל אחת מהשורות ב- main קיבעו האם תיוצר שגיאת קומפילציה או שגיאת זמן ריצה. אם לא, רשמו בדף התשובות מה תדפיס התוכנית. אם כן, רשמו בדף התשובות את סוג השגיאה והניחוי שהשורה נמחקה מה- main.

```
public class A {

    public A () { System.out.print("A"); }
    public void arik () { System.out.print("Arik_A"); }
    public void yosef () { arik(); }
}

public class B extends A {

    public B () { System.out.print("B"); }
    public void arik () { System.out.print("Arik_B"); }
    public void yosef () { System.out.print("Yosef"); }
    public void superYosef () { super.yosef(); }
}

public class What {

    public static void main(String[] args) {

        ( 1) A a = (A) new B();
        ( 2) A aa = new A();
        ( 3) A ab = new B();
        ( 4) B ba = new A();
        ( 5) B bb = new B();

        ( 6) aa.yosef();
        ( 7) ab.yosef();
        ( 8) bb.yosef();

        ( 9) ((A) aa).yosef();
        (10) ((A) bb).yosef();
        (11) ((A) bb).superYosef();

        (12) ((B) aa).yosef();
        (13) ((B) aa).superYosef();
        (14) ((B) ab).superYosef();
        (15) ((B) bb).superYosef();
    }
}
```


מבוא למדעי המחשב
202-1-101-1

סמסטר ב', תשס"ח, 2007/8
מבחן

דף תשובות

שאלה 1 סעיף א (10 נקודות)

```
public static boolean isPerfect (int [] vec) {  
  
    boolean ans = true;  
  
    for (int i = 0 ; i < vec.length && ans ; i++){  
        if ((vec[i] > vec.length) || (vec[i] < 1))  
            ans = false;  
        for (int j = i+1 ; j < vec.length && ans ; j++){  
            if (vec[i] == vec[j])  
                ans = false;  
        }  
    }  
  
    return ans;  
}
```

שאלה 1 סעיף ב (10 נקודות)

```
public static boolean isInclude(int[] vec, int u) {  
  
    int v = 1;  
  
    for (int i = 0 ; i < vec.length && v<=u ; i++){  
        if (vec[i] == v*v)  
            v++;  
    }  
  
    return v > u;  
}
```

שאלה 2 סעיף א (10 נקודות)

```
private static int where(int[] vec, int l, int p, int r) {  
    if (p > r )  
        return -1;  
    int suml = sum(vec,l,p-1);  
    int sumr = sum(vec,p,r);  
    if (suml == sumr)  
        return p;  
    else  
        return where(vec,l,p+1,r);  
}
```

שאלה 2 סעיף ב (10 נקודות)

```
private static boolean isBalanced(int[] vec, int l, int r,  
int k) {  
    if (k==0)  
        return true;  
    int pos = where(vec,l,l,r);  
    if (pos == -1)  
        return false;  
    else  
        return (is(vec,l,pos-1,k-1) && is(vec,pos,r,k-1));  
}
```

שאלה 3 סעיף א (10 נקודות)

```
public boolean equals (Nat other) {  
    if (isZero() | other.isZero())  
        return isZero() == other.isZero();  
    return pre.equals(other.pre);  
}
```

שאלה 3 סעיף ב (10 נקודות)

```
public Nat plus (Nat other) {  
    if (other.isZero())  
        return this;  
    return succ().plus(other.pre);  
}
```

שאלה 4 סעיף א (10 נקודות)

```
public class IterBSTree extends BSTree implements Iter {

    private Node current;

    public void reset() {
        current = root;
        if (current != null)
            while (current.left != null)
                current = current.left;
    }

    public boolean hasNext() {
        return current != null;
    }

    public Object next() {

        Object x = current.data;

        if (current.right != null) {

            current = current.right;
            while (current.left != null)
                current = current.left;
        } else {

            while (current.parent != null && current.parent.left != current)
                current = current.parent;
            current = current.parent;
        }

        return x;
    }
}
```

שאלה 4 סעיף ב (15 נקודות)

```
public Integer kmax(int k) {  
    return kmax(root, k);  
}  
  
private Integer kmax(Node p, int k) {  
    int n = p.size;  
  
    if (p.right != null)  
        n = n - p.right.size;  
  
    if (n == k)  
        return p.data;  
  
    if (k < n)  
        return kmax(p.left, k);  
  
    return kmax(p.right, k-n);  
}
```

שאלה 5 (15 נקודות)

- AB .1
- A .2
- AB .3
- Compilation error, Incompatible types .4
- AB .5
- Arik_A .6
- Yosef .7
- Yosef .8
- Arik_A .9
- Yosef .10
- Compilation Error, cannot find symbol superYosef .11
- Runtime Error, ClassCastException .12
- Runtime Error, ClassCastException .13
- Arik_B .14
- Arik_B .15