



מס' שאלון - 458

9

ביולי 2014

סמסטר 2014ב

מס' מועד 85

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 14 עמודים

מבנה הבחינה:

- * קראו בעיון את ההנחיות שלהלן:
- * בבחינה יש שש שאלות.
- * כל התכניות צריכות להיות מתועדות היטב.
- * יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
- * יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית.
- * תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85 % מהניקוד.
- * יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
- * תכנית לא יעילה לא תקבל את מלוא הנקודות.
- * אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את המחלקה למחברת הבחינה. מספיק להפנות למקום הנכון, ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').
- * אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.
- * יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהציון.
- * בכתיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה
- * אין להשתמש במשתנים גלובליים!
- * את התשובות לשאלות 3 - 6 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!
- * אפשר לתעד בעברית. אין צורך בתיעוד API.

חומר עזר:

ספר הלימוד : java software solutions מאת : lewis/loftus
חוברת השקפים של הקורס.
מותרות הערות בכתב יד, ע"ג הספרים.
אין להכניס חומר מודפס או כל חומר אחר מכל סוג שהוא.

בהצלחה !!!

החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתוך מחברת התשובות



חלק א – עליכם לענות על כל השאלות בחלק זה במחברת הבחינה

שאלה 1 - 25 נקודות

עבור מערך דו-ממדי של מספרים שלמים, נגדיר **תולעת** (worm) באורך k במערך להיות סדרה של k תאים, שכל אחד מהם סמוך לקודמו, כלומר, הוא אחד מארבעת שכניו במערך (שכן מעליו, מתחתיו, מימינו ומשמאלו, אך לא באלכסון לו), כך שהערכים בתאים מהווים סדרה עולה של מספרים שלמים עוקבים $n, n+1, \dots, n+k-1$.
 n , האיבר הראשון בסדרה, עשוי להיות כלשהו.
למשל, במערך:

	0	1	2	3	4
0	3	13	15	28	30
1	50	51	52	29	30
2	51	10	53	54	55
3	53	12	14	53	11

- בתא $[0][3]$ מתחילה התולעת באורך שלוש הבאה: 28, 29, 30 (הנגמרת בתא $[1][4]$)
 - בתא $[1][0]$ מתחילה התולעת באורך שש הבאה: 50, 51, 52, 53, 54, 55 (הנגמרת בתא $[2][4]$)
 - בתא $[1][0]$ מתחילה התולעת באורך שתיים הבאה: 50, 51 (הנגמרת בתא $[2][0]$)
- כתבו שיטה סטטית **רקורסיבית** המקבלת מערך דו-ממדי, ומחזירה את אורכה של התולעת הארוכה ביותר במערך.
לדוגמא, אם השיטה תקבל את המערך שלעיל, היא תחזיר את הערך 6.

חתימת השיטה:

```
public static int longestWorm (int [][] mat)
```

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).

מותר לשנות את המערך במהלך השיטה, אבל חובה להחזיר אותו למצבו ההתחלתי בסופה.

שאלה 2 - 25 נקודות

נתונות השיטות הסטטיות הבאות:

```
public static boolean what (int n)
{
    return (f(n)%2 == 0);
}

public static int f (int n)
{
    int c = 1;
    for (int i=2; i<=n; i++)
    {
        if (n%i == 0)
            c++;
    }
    return c;
}
```

- א. מה מבצעת השיטה what בהינתן לה מספר שלם חיובי n כלשהו. הסבירו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת. כלומר, כתבו במפורש מה המשמעות של הערך המוחזר מהשיטה what. (4 נק').
- ב. מהי סיבוכיות הזמן של השיטה what ? (2 נק')
- ג. כתבו את השיטה what כך שתבצע את מה שביצעה בסעיף א בסיבוכיות זמן ריצה קטנה יותר. (17 נק')

שימו לב:

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

- ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (2 נק')

אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3-6 יש לכתוב על גבי השאלון.
לא נבדוק תשובות שייכתבו במקום אחר!**

שאלה 3 - 12 נקודות

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int num)
    {
        _number = num;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.
בין השיטות נתונות השיטות הבאות:

```
public static boolean f1 (Node t)
{
    if (t==null)
        return true;
    if (isLeaf(t))
        return true;
    if (t.getRightSon()== null)
        return (t.getNumber() < t.getLeftSon().getNumber())
            && f1(t.getLeftSon());
    if (t.getLeftSon()== null)
        return (t.getNumber() < t.getRightSon().getNumber())
            && f1(t.getRightSon());
    return (t.getNumber() < t.getLeftSon().getNumber())
        && (t.getNumber() < t.getRightSon().getNumber())
        && f1(t.getLeftSon())
        && f1(t.getRightSon());
}
```



```
public static boolean f2 (Node t)
{
    if (t==null)
        return true;
    if (isLeaf(t))
        return true;
    if (t.getRightSon()== null)
        return (t.getNumber() < t.getLeftSon().getNumber())
                && f2(t.getLeftSon());
    if (t.getLeftSon()== null)
        return (t.getNumber() < t.getRightSon().getNumber())
                && f2(t.getRightSon());
    return ((t.getNumber() < t.getLeftSon().getNumber())
            && f2(t.getLeftSon()))
           || ((t.getNumber() < t.getRightSon().getNumber())
              && f2(t.getRightSon()));
}

public static boolean isLeaf (Node t)
{
    return (t.getLeftSon() == null) &&
           (t.getRightSon() == null);
}
```

ענו על הסעיפים הבאים:

3) נקי (א) ציירו עץ בן 9 צמתים לפחות, בעל 4 רמות לפחות, שאם נעביר אותו כפרמטר לשיטה f1, השיטה תחזיר את הערך true.

התשובה היא:

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____



3 נק') (ב) ציירו עץ בן 9 צמתים לפחות, בעל 4 רמות לפחות, שאם נעביר אותו כפרמטר לשיטה f2, השיטה תחזיר את הערך true.

התשובה היא:

<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

3 נק') (ג) ציירו עץ בן 9 צמתים לפחות, בעל 4 רמות לפחות, שאם נעביר אותו כפרמטר לשיטה f1, השיטה תחזיר את הערך false, ואם נעביר אותו כפרמטר לשיטה f2 השיטה תחזיר true.

התשובה היא:

<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

3 נק') (ד) ציירו עץ בן 9 צמתים לפחות שהם שונים זה מזה, בעל 4 רמות לפחות, שאם נעביר אותו כפרמטר לשיטה f1, השיטה תחזיר את הערך false, ואם נעביר אותו כפרמטר לשיטה f2 השיטה תחזיר false.

התשובה היא:

<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

שאלה 4 - 12 נקודות

נתונות המחלקות Base, Sub ו-SubSub הבאות (באותה חבילה):

```
public class Base
{
    protected int _i = 0;

    public Base (int i)
    {
        _i = i;
    }

    public Base (Base b)
    {
        _i = b._i;
    }

    public Base foo ()
    {
        return new Base (this);
    }
}

//-----//

public class Sub extends Base
{
    public Sub (int i)
    {
        super (i);
    }

    public Sub (Sub s)
    {
        super (s._i * 2);
    }

    public Base foo ()
    {
        return this;
    }

    public boolean equals (Object o)
    {
        return ((Sub)o)._i == _i;
    }
}
```



```

public class SubSub extends Sub
{
    public SubSub()
    {
        super (1);
    }

    public boolean equals (SubSub s)
    {
        return (s!= null) && (_i == s._i);
    }

    public boolean equals (Object o)
    {
        return (o!= null) && (o instanceof SubSub)
            && (_i == ((SubSub) o)._i);
    }

    public static void main (String [] args)
    {
        Base b1 = new Base(1);
        Base b2 = b1;
        Base b3 = b2.foo();
        Base b4 = new Sub(1);
        Base b5 = b4.foo();
        Sub s1 = new Sub(1);
        Sub s2 = new SubSub();

        // HERE
    }
}

```

עבור כל אחת משורות הקוד הבאות בדקו האם ניתן להגדירה בשיטה main של המחלקה Sub במקום ההערה //HERE. סמנו את האפשרות המתאימה: האם תהיה שגיאת קומפילציה (אם כן, ציינו מה השגיאה), או שהתכנית תעוף בזמן הריצה (אם כן, ציינו מאיזו סיבה), או שהקוד ירוץ בצורה תקינה (אם כן, ציינו מה יהיה פלט ההדפסה). נמקו בקצרה.

1. System.out.println (b1 == b3);

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

2. System.out.println (b1.equals(b3));

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

System.out.println (b1.equals(b4)); .3

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (b4.equals(b1)); .4

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (b5 == b4); .5

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (b5.equals(b4)); .6

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (s1 == b4); .7

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (s1.equals(b4)); .8

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (s2.equals(b4)); .9

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

✓

System.out.println (s1 == s2); .10

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

System.out.println (s1.equals(s2)); .11

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

System.out.println (b4.equals(s1)); .12

שגיאת קומפילציה (מהי?) / שגיאה בזמן ריצה (מה הסיבה?) / קוד תקין (מה יודפס?)

שאלה 5 - 16 נקודות

נתונה המחלקה IntNode הבאה, המייצגת איבר ברשימה:

```
public class IntNode {  
    private int _value;  
    private IntNode _next;  
  
    public IntNode(int val, IntNode n) {  
        _value = val;  
        _next = n;  
    }  
  
    public int getValue() {  
        return _value;  
    }  
    public IntNode getNext() {  
        return _next;  
    }  
  
    public void setValue(int v) {  
        _value = v;  
    }  
    public void setNext(IntNode node) {  
        _next = node;  
    }  
}
```


נתונה רשימה מקושרת של מספרים שלמים, הממומשת בעזרת המחלקה IntList שלהלן:

```
public class IntList
{
    private IntNode _head;

    public IntList( ) {
        _head = null;
    }

    public boolean f(int num)
    {
        IntNode p=_head;
        while (p != null)
        {
            if (p.getValue() == num)
                return true;
            p = p.getNext();
        }
        return false;
    }

    public int g( int num)
    {
        int t1 = num;
        int t2 = num;

        while (f(t1))
            t1++;
        while (f(t2))
            t2--;

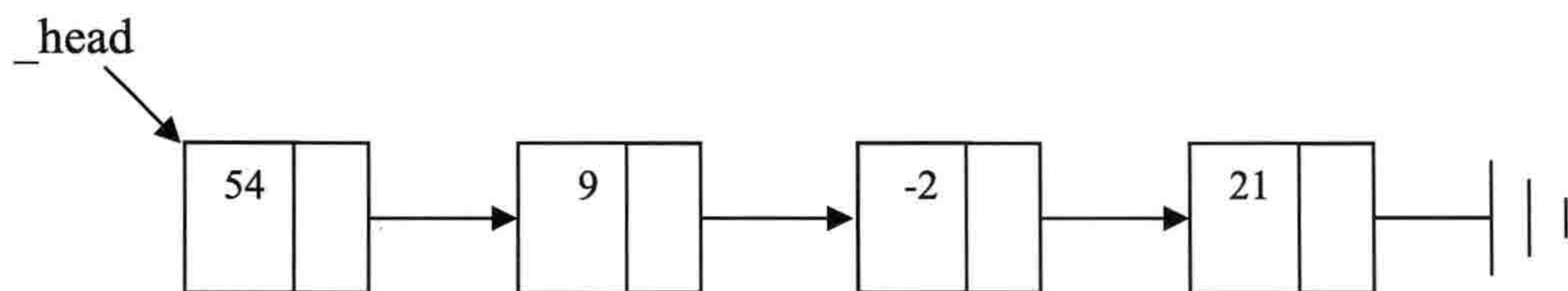
        return (t1-num) + (num-t2) -1;
    }

    public int what()
    {
        int m = 0;
        IntNode p=_head;
        while (p != null)
        {
            int res = g(p.getValue());
            if (res > m)
                m = res;
            p = p.getNext();
        }
        return m;
    }

    . . .          // other methods
}
```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים, ואורכה הוא n.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.
 כך לדוגמא, נסמן { 54 , 9 , -2 , 21 } את הרשימה שלהלן:



סעיף א (8 נקודות)

הקיפו בעיגול את כל הטענות הנכונות שכתובות להלן:

- אם נפעיל את השיטה g על הרשימה { 9, 7, 2, 8, 3 } והיחיד $num = 4$, נקבל את התשובה 2.
- אם נפעיל את השיטה g פעם אחת על הרשימה { 9, 7, 2, 8, 3 } והמספר $num = 8$ ופעם שניה על הרשימה { 2, 3, 7, 8, 9 } והמספר $num = 7$ נקבל אותה תשובה.
- אם נפעיל את השיטה what פעם אחת על הרשימה { 2, 7, 2, -8, 3 } ופעם שניה על הרשימה { 12, 3, -1, 5, 9, -3, 16, 7 } נקבל אותה תשובה.
- אם נפעיל את השיטה what פעם אחת על הרשימה { 2, 5, -8, 1, -1, 7, -3, 6, -2 } ופעם שניה על הרשימה { 10, 7, 3, -1, 5, 9, -3, 8 } נקבל אותה תשובה.

סעיף ב (8 נקודות)

מה מבצעת השיטה what באופן כללי? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת.

שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה what, כשהיא מופעלת על רשימה כלשהי.

—
—
—

✓

שאלה 6-10 נקודות

נתונות השיטות הסטטיות הבאות:

```
public static void humpty (int []a, int low, int high)
{
    int x=low;
    if (low >= high)
        return;
    for (int i=low; i<=high; i++)
        if (a[i] < a[x])
            x = i;
    if (x!= low)
    {
        int temp = a[low];
        a[low] = a[x];
        a[x] = temp;
    }
    dumpty (a, low, high);
}

public static void dumpty (int []a, int low, int high)
{
    int x=low;
    if (low >= high)
        return;
    for (int i=low; i<=high; i++)
        if (a[i] >= a[x])
            x = i;
    if (x!= high)
    {
        int temp = a[high];
        a[high] = a[x];
        a[x] = temp;
    }
    humpty (a, low+1, high-1);
}
```


ענו על שלושת הסעיפים הבאים:

(2 נק') (i) אם נתון המערך a הבא:

0	1	2	3	4	5
4	5	3	2	7	6

איך יראה המערך a לאחר הקריאה לשיטה $\text{humpty}(a, 0, 5)$, ללא ביצוע השורה האחרונה ($\text{dumpty} \dots$)

התשובה היא:

0	1	2	3	4	5

(2 נק') (ii)

איך יראה המערך a שהתקבל מסעיף i לאחר הקריאה לשיטה $\text{dumpty}(a, 0, 5)$, ללא ביצוע השורה האחרונה ($\text{humpty} \dots$)

התשובה היא:

0	1	2	3	4	5

(6 נק') (iii) מה מבצעות השיטות $\text{humpty} - \text{dumpty}$ באופן כללי בהינתן להן כפרמטרים מערך מלא במספרים שלמים, ושני אינדקסים שלמים – התא הראשון והתא האחרון במערך? שימו לב, עליכם לתת תיאור ממצה של מה עושות השיטות באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. תנו תיאור ממצה למה שמתקבל לאחר הרצת השיטה humpty ואשר פונה לשיטה dumpty שזו פונה בחזרה לשיטה humpty וכן הלאה.

אין צורך להפריד בין השיטות, אלא לכתוב מה מתקבל בסך הכל.

התשובה היא

-

-

בהצלחה