

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 12 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.

החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתוך מחברת התשובות

בהצלחה !!!



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

שאלה 1 - 25 נקודות

נתון מערך דו-ממדי `mat` המכיל מספרים שלמים (חיובים, שליליים ואפסים), ומספר שלם `number`. אנחנו מעוניינים לעבור במסלול במערך מהתא הראשון $(0,0)$ עד לתא האחרון $(mat.length-1, mat[0].length-1)$, כאשר אפשר לעבור מתא (i, j) אך ורק שורה אחת למטה או עמודה אחת ימינה, כלומר לאחד משני התאים $(i+1, j)$ או $(i, j+1)$. בכל ביקור בתא, המספר שנמצא בתא מתווסף (או מופחת) למספר `number`. אפשר לבקר בתא רק אם המספר `number` הוא חיובי ממש (לפני ואחרי שהתא מתווסף למספר).

כתבו שיטה סטטית רקורסיבית המקבלת מערך דו-ממדי `mat` המלא במספרים שלמים, ומחזירה את ערכו של המספר המינימלי שצריך להיות בהתחלה ב-`number` כדי להצליח לעבור מהתא הראשון לתא האחרון במערך.

לדוגמא, אם המערך הוא:

	0	1	2
0	-2	-3	3
1	-5	-10	1
2	10	30	-5

אז המספר המינימלי ההתחלתי של `number` צריך להיות 7. שכן, אז אפשר ללכת במסלול הזה:

$$(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (2,2)$$

אם בהתחלה `number = 7`, לאחר ביקור בתא $(0,0)$ ערכו יהיה 5, לאחר ביקור בתא $(0,1)$ ערכו יהיה 2, לאחר ביקור בתא $(0,2)$ ערכו יהיה 5, לאחר ביקור בתא $(1,2)$ ערכו יהיה 6 ולאחר ביקור בתא $(2,2)$ ערכו יהיה 1. כלומר, הערך של `number` היה חיובי לכל אורך המסלול.

שימו לב, אם היינו הולכים במסלול אחר $(0,0) \rightarrow (1,0) \rightarrow (2,0) \rightarrow (2,1) \rightarrow (2,2)$, היינו צריכים ערך של לפחות 8 כדי ש-`number` תמיד יהיה חיובי.

חתימת השיטה היא:

```
public static int minPoints(int [][] m)
```

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).

אל תשכחו לתעד את מה שכתבתם.

שאלה 2 - 25 נקודות

מערך נקרא "ממוין לסירוגין" אם שני התת-מערכים שלו המורכבים אחד ממספרים המופיעים באינדקסים הזוגיים (0, 2, ...) והשני ממספרים המופיעים באינדקסים האי-זוגיים (1, 3, ...) הם ממוינים כל אחד.

במלים אחרות: מערך נקרא "ממוין לסירוגין" אם ניתן לייצר ממנו שני תת-מערכים ממוינים שאחד מורכב מהמספרים המופיעים באינדקסים הזוגיים והשני מורכב מהמספרים המופיעים באינדקסים האי-זוגיים.

לדוגמא, המערך הבא ממוין לסירוגין

0	1	2	3	4	5	6
1	2	5	3	6	10	9

בעוד שהמערך הבא אינו ממוין לסירוגין (כי 4 מופיע אחרי 5 בתת-מערך הזוגיים).

0	1	2	3	4	5	6
1	2	5	3	4	10	9

כתבו שיטה findX המקבלת מערך ממוין לסירוגין של מספרים שלמים a ומספר שלם x ומחזירה true אם קיימים ב-a שני איברים סמוכים שסכומם הוא x, ו-false אחרת.

לדוגמא, עבור המערך

0	1	2	3	4	5	6
1	2	5	3	6	10	9

כאשר $x=9$, השיטה תחזיר true (שני האיברים הסמוכים 3 ו-6 סכומם 9) ועבור אותו מערך a כאשר $x=5$ השיטה תחזיר false. (אין שני איברים סמוכים שסכומם 5). יש להתייחס גם למקרה בו במערך יש רק איבר אחד, ואז השיטה צריכה להחזיר false.

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. אפשר להניח שהמערך מלא בנתונים ושהוא ממוין לסירוגין ואין צורך לבדוק זאת.

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 - 10 נקודות

נניח שהמחלקה Node שלהלן מממשת עץ חיפוש בינרי (Binary Search Tree).

```
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; }
}
```

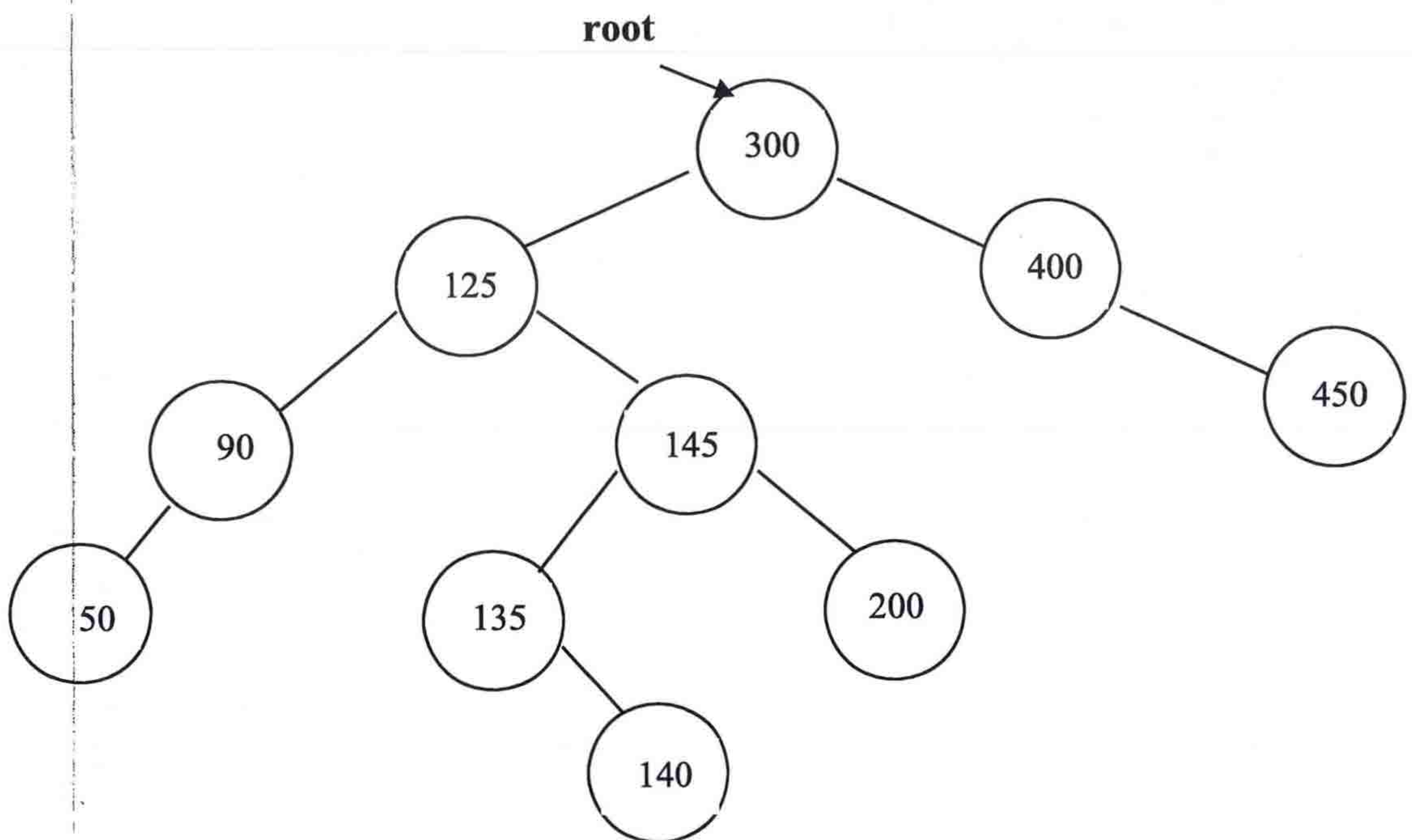
המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.

בין השיטות נתונה השיטות f ו-g הבאות:

```
public static int f(Node t) {
    if(t == null)
        return 0;
    return f(t.getLeftSon()) + f(t.getRightSon()) + 1;
}

public static int g(Node t, int x)
{
    if(t == null)
        return 0;
    if(t.getNumber() > x)
        return g(t.getLeftSon(), x);
    if(t.getNumber() < x)
        return f(t.getLeftSon()) + 1 + g(t.getRightSon(), x);
    return f(t.getLeftSon());
}
```


נתון עץ חיפוש בינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

2 נק' (i) מה מבצעת השיטה f באופן כללי? 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, מה המשמעות של הערך המוחזר מהקריאה לשיטה f.

התשובה היא:

השיטה f מחזירה את מספר הענפים שממנו נובע הערך 200.

3 נק' (ii) איזה ערך תחזיר השיטה g בעקבות הקריאה BinaryTree.g(root, 200)?

התשובה היא:

6.

5 נק' (iii) מה מבצעת השיטה g באופן כללי בהינתן לה שורש של עץ חיפוש בינרי ומספר x שלם כלשהו? 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, מה המשמעות של הערך המוחזר מהקריאה לשיטה g בהתייחס ל-root ו-x. התייחסו למקרי הקצה השונים.

התשובה היא:

השיטה g מחזירה את מספר הענפים שממנו נובע הערך x.

שאלה 4 - 15 נקודות

נתונה המחלקה הבאה:

```
public abstract class A
{
    private int _num;
    public A() {
        _num = 2;
    }

    public int getNum() {
        return _num;
    }

    public abstract boolean f(int x);
}
```

בכל אחד מהסעיפים הבאים נתונה הגדרה של מחלקה B היורשת מ-A. בכל סעיף כתבו האם קוד המחלקה תקין, או אם היא לא עוברת קומפילציה הסבירו למה. כל סעיף שווה 3 נקודות.

1.

```
public class B extends A {
    public boolean f(double x)
    { return x == 2.0; }
}
```

התשובה היא לא עוברת קומפילציה, כי B לא מיישמת את השיטה f של A, קוצצת העדשה ולא בריסה (מימוש חלקי).

2.

```
public abstract class B extends A {
    public boolean f(int x)
    { return x == 2; }
}
```

התשובה היא תקין.

התשובה היא שגוה קומפילציה, אין גנאי סיכור
לא בביטוי, יש לו רק גנאי רץ.

4.

```
public abstract class B extends A {
    public B(int n) {
        n += getNum();
        System.out.println(n);
    }
}
```

התשובה היא נכון.

5.

```
public class B extends A {
    public boolean f(int x) {
        Object obj1 = new A();
        (A)obj1.f(x);
    }
}
```

התשובה היא שגוה קומפילציה, לא כן, ליצור אובייקט
של A כי A אבסטרקט.

שאלה 5 - 13 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:
 השיטה what פועלת על רשימות שהן ממוינות בסדר לא יורד. (אשר).

```

public class IntList
{
    private IntNode _head;

    public IntList( ) {
        _head = null;
    }

    public double what (IntList other){
        int c = 0;
        IntNode cu = _head;
        while ( cu != null ) {
            cu = cu.getNext();
            c++;
        }
        cu = other._head;
        while ( cu != null ) {
            cu = cu.getNext();
            c++;
        }
        IntNode cu1 = _head;
        IntNode cu2 = other._head;
        int m=c/2;
        if(c%2==1)
            m++;
        int count=1;
        while (cu1!=null && cu2!=null)
        {
            if (count==m )
                if(c%2==1)
                    if (cu1.getValue()<cu2.getValue())
                        return cu1.getValue();
                    else
                        return cu2.getValue();
                else
                    if (cu1.getValue()<cu2.getValue())
                        if (cu1.getNext()!=null&&cu1.getNext().getValue()<
                            cu2.getValue())
                            return (cu1.getValue()+cu1.getNext().getValue())/2.0;

```



```

        else
            return (cu1.getValue()+cu2.getValue())/2.0;
        else
            if (cu2.getNext()!=null&&cu2.getNext().getValue() <
                cu1.getValue())
                return (cu2.getValue()+cu2.getNext().getValue())/2.0;
            else
                return (cu1.getValue()+cu2.getValue())/2.0;

        if (cu1.getValue() < cu2.getValue())
            cu1=cu1.getNext();
        else
            cu2=cu2.getNext();
        count++;
    } //end of while

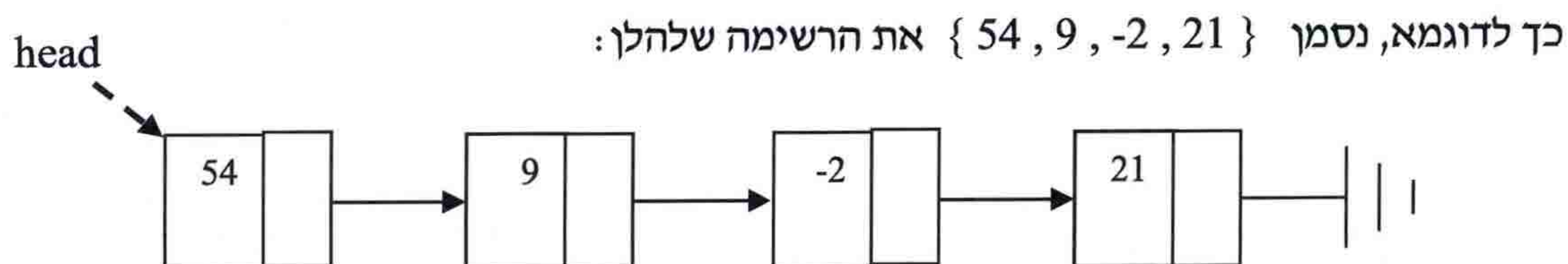
    IntNode newCu;
    if (cu1!=null)
        newCu=cu1;
    else
        newCu=cu2;
    while (count!=m) {
        newCu= newCu.getNext();
        count++;
    }
    if (c%2==1)
        return newCu.getValue();
    else
        return (newCu.getValue()+ newCu.getNext().getValue())/2.0;
} //end of method what

} //of class IntList

```

הניחו שיש במחלקה גם שיטה שמכניסה ערכים בצורה ממוינת לרשימה. אינכם צריכים לדאוג לכך. אתם יכולים גם להניח שהרשימות אינן ריקות.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.



סעיף א (3 נקודות)

אם נפעיל את השיטה **what** על הרשימה $\{1, 3, 5, 6\}$ והרשימה **other** היא $\{2, 2, 4\}$ מה יוחזר?

3.

סעיף ב (3 נקודות)

אם נפעיל את השיטה **what** על הרשימה $\{1, 3, 5, 6\}$ והרשימה **other** היא $\{8, 12, 14, 16\}$ מה יוחזר?

סעיף ג (2 נקודות)

אם נפעיל את השיטה **what** על הרשימה $\{8, 12, 14, 16\}$ והרשימה **other** היא $\{1, 3, 5, 6\}$ מה יוחזר?

סעיף ד (5 נקודות)

מה מבצעת השיטה **what** באופן כללי כשהיא מופעלת על רשימה ממוינת ומקבלת כפרמטר רשימה ממוינת אחרת? 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. עליכם לכתוב מה המשמעות של הערך המוחזר מהשיטה. שימו לב למקרי הקצה.

שאלה 6-12 נקודות

נתונה השיטות הסטטיות הבאות:

```
public static int g(String s, int i)
{
    int mg=0;
    for (int j=1; j<=i; j = j+1)
    {
        if (s.charAt(j) < s.charAt(mg))
            mg=j;
    }
    return mg;
}

public static String f(String s,int i)
{
    String r ="";
    for(int j=0;j<=i;j++)
        r +=s.charAt(i-j);
    r +=s.substring(i+1);
    return r ;
}

public static String what(String s)
{
    for (int i=s.length()-1; i>0; i--)
    {
        int m = g(s,i);
        s=f(s,m);
        s=f(s,i);
    }
    return s;
}
```

להזכירכם:

- השיטה `length()` מחזירה את אורך המחרוזת עליה היא פועלת.
לדוגמא, אם `s = "abcdef"` אז אורכה הוא 6
- השיטה `charAt(i)` מחזירה את התו ה- `i` במחרוזת עליה היא פועלת
לדוגמא, אם `s = "abcdef"` ו- `i=2`, אז התו שיוחזר הוא 'c'
- השיטה `substring(i)` מחזירה את התת-מחרוזת החל במקום ה- `i` ועד לסוף המחרוזת עליה היא פועלת.
לדוגמא, אם `s = "abcdef"` ו- `i=2`, אז התת-מחרוזת שתוחזר היא "cdef".

ענו על שלושת הסעיפים הבאים:

(3 נק') (i) מה מבצעת השיטה g באופן כללי בהינתן לה מחרוזת תווים s ומספר שלם i כלשהו? 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. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות, וכן שהפרמטר i מקיים $0 \leq i < s.length()$.

התשובה היא:

השיטה מחזירה את האינדקס של ה-1 ביגור כתג מחרוזת
שכן תייל המחרוזת לא ה'.

(4 נק') (ii) מה מבצעת השיטה f באופן כללי בהינתן לה מחרוזת תווים s ומספר שלם i כלשהו? 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. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות, וכן שהפרמטר i מקיים $0 \leq i < s.length()$.

התשובה היא:

השיטה מחזירה מחרוזת חזרה בה הגי' י' (כולל) הם הכוכים
בסדרם מיוחדים הקודם (לא) הגי' אחרי ה' נשאר במקום כמו במחרוזת
המקורית.

(5 נק') (iii) מה מבצעת השיטה what באופן כללי בהינתן לה מחרוזת תווים s? 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. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות.

התשובה היא:

השיטה מחזירה מחרוזת מהווינת בסדר יורד.
הג ההמשך יהיה הצדל ביגור והאחרון יהיה הקטן ביותר.

בהצלחה