

477 מס' שאלון -

24

י"ד בתמוז תשפ"א

60 מס' מועד

ביוני 2021

סמסטר 2021ב

20441 / 4

שאלון בבחינת גמר

Java - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 12 עמודים

מבנה הבחינה:

קראו בעיון את הנקודות של להלן:

* בבחינה יש חמישה שאלות.

* כל התכניות צריכה להיות מתועדות היטב.

יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.

יש לבחור בשמות משתנים למשתנים, לפונקציות ולקבועים שבתכנית.

תכנית שלא תتوודכן נדרש לעיל לקבל לכל היתר % 85 מהניקוד.

* יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.

תכנית לא יעילה לא תקבל את מלאה הנקודות.

* אם ברצונכם להשתמש בתשובותם בשיטה או בחלוקת הכתיבה בחוברת השקפים,

אין צורך שתעתיקו את השיטה או את המחלוקת למחברת הבחינה. מספיק להפנות

למקום הנכון, וב惟ד שהפניה תהיה מדוקנית (פרמטרים, מקום וכו').

* אין להשתמש בחלוקת קיימות ב- Java, חוץ מalto המפורטות בשאלות הבחינה.

* יש לשמור על סדר; תכנית הכתיבה בצורה בלתי מסודרת עלולה לגרום מהצין.

* בכתיבה התכניות יש להשתמש אר ורך במרכיבי השפה שנלמדו בקורס זה

אין להשתמש במשתנים גלובליים!

* אפשר לטעות בעברית. אין צורך בתיעוד API.

את התשובות לשאלות 3-5 יש לכתוב על גבי השאלון בלבד.

לא יבדקו תשבות שייכתו במקום אחר.

חומר עזר:

חוברות השקפים 1-6, 7-12.

אין להכניס חומר מודפס נוסף או חומר אחר מכל סוג.

בהצלחה !!!

הצדiron

למשגיח את השאלון

וכל עזר אחר ע

S100407771



חלק א - עלייכם לענות על כל השאלות בחלק זה במחברת הבחינה

שאלה 1 (25 נקודות)

נתון מערך חד-ממדי `arr` שערכוו הם מספרים שלמים חיוביים ממש. (אין צורך לבדוק זאת). כתבו שיטה סטטית בוליאנית רקורסיבית המחזירה `true` אם ניתן לחלק את הערכים שבמערך לשולש קבוצות שוונות שלכלן סכום זהה, ו- `false` אחרת. שימו לב:

- המספרים בכל קבוצה לאו דווקא נמצאים בתאים צמודים זה לזה.
- לא יתכן שתא אחד נמצא ביותר ממספר קבוצה אחת מהשלוש.
- כל אחד מהמספרים במערך צריך להיות באחת משלוש הקבוצות.
- גודלו של שלוש הקבוצות לא צריך להיות זהה, אלא הסכומים שלהם.

לדוגמא,

אם המערך הוא $\{5, 3, 5, 2, 1, 2, 4, 7\}$

אזי השיטה תחזיר `true` שכן אפשר לחלק את איברי המערך כך:

$$8 + 2 = 4 + 5 + 1 = 3 + 7 = 10$$

אם המערך הוא $\{5, 3, 5, 2, 1, 2, 4, 7\}$

אזי השיטה תחזיר `false` שכן אין אפשרות לחלק את איברי המערך לשולש קבוצות שסכוםן זהה.

התימת השיטה היא:

```
public static boolean split3 (int [] arr)
```

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

אפשר להשתמש בעומסת-יתר (overloading). אפשר להניח שהמערך אינו `null` וaino ריק.

אין לשנות את תוכן המערך (אפילו לא זמנית), ולא להשתמש במערך עזר.

אין צורך לדאוג ליעילות השיטה! אבל כמובן צריך לשימוש לב לא לעשות קריאות רקורסיביות מיותרות!

אל תשחחו לטעד את מה שתכתבם!

```
public static boolean split3(int [] arr) {
    return split3(arr,0,0,0,0);
}

private static boolean split3(int [] arr,int i,int sum1,int sum2,int sum3) {

    if(arr.length < 3)
        return false;

    if(i == arr.length) {
        if(sum1 == sum2 && sum2 == sum3 && sum1 == sum3) {
            return true;
        }else {
            return false;
        }
    }

    boolean way1 = split3(arr,i+1,sum1+arr[i],sum2,sum3);
    boolean way2 = split3(arr,i+1,sum1,sum2+arr[i],sum3);
    boolean way3 = split3(arr,i+1,sum1,sum2,sum3+arr[i]);
    return way1 || way2 || way3;
}
```

שאלה 2 (25 נקודות)

נתון מערך a המלא במספרים שלמים אי-שליליים (אין צורך לבדוק זאת). כתבו שיטה סטטית המקבלת את המערך כפרמטר וכן מספר שלם k . השיטה צריכה למצוא את התת-מערך (sub-array) הקטן ביותר שהסכום של איבריו גדול מהמספר k , ולהחזיר את האורץ של התת-מערך זהה. אם אין תת-מערך כזה (כלומר, הסכום של כל איברי המערך עדין קטן ממהספר k), השיטה תחזיר את הערך $a.length+1$ (אורך המערך a ועוד 1). **להזכירכם – בתת-מערך התאים סמוכים זה לזה (רצופים).**

חתימת השיטה היא:

```
public static int smallestSub (int []a, int k)
```

דוגמאות:

אם המערך a הוא

0	1	2	3	4	5
1	4	13	6	0	19

- אם $k = 22$ השיטה תחזיר את הערך 3 כי התת-מערך הקטן ביותר הוא $\{6, 13, 6\}$ או התת-מערך $\{6, 0, 19\}$.
- אם $k = 42$ השיטה תחזיר את הערך 6 כי התת-מערך הקטן ביותר הוא המערך כולו.
- אם $k = 43$ השיטה תחזיר את הערך 7 כי אין תת-מערך שסכום איבריו גדול מ- 43.
- אם $k = 26$ השיטה תחזיר את הערך 4 כי התת-מערך הקטן ביותר הוא $\{13, 6, 0, 19\}$.
- אם $k = 2$ השיטה תחזיר את הערך 1 כי הסכום של התת-מערך $\{4\}$ גדול מ- 2 (וכך גם לגבי התת-מערכות $\{13\}$ או $\{6\}$ או $\{19\}$).

עליכם להחזיר את אורך התת-מערך הקטן ביותר ולא את איברי התת-מערך הזה.

מה סיבות זמן הריצה והמקום של השיטה שתכתבם? הסבירו תשובהכם.

שימוש לב:

השיטה שתכתבו צריכה להיות עיליה כלל הניתן, גם מבחינית סיבות זמן וגם מבחינית סיבות מקום. תשובה שאינה עיליה מספיק כלומר, שתהייה בסיבות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

אל תשחחו לטעד את מה שתכתבם!

```
public static int smallestSub(int [] a, int k) {
    int i=0;
    int j=0;
    int sum=0;
    int smallest = a.length-1;

    while(i < a.length)
    {
        sum+=a[i]; // count all the index's in the array.
        while (sum > k) // if sum is bigger than k
        {
            smallest = i-j+1;
            sum -=a[j];
            j++;
        }
        i++;
    }
    return smallest;
}
```

**חלק ב - את התשובות לשאלות 3- 5 יש לכתוב על גבי השאלה!
לא נבדוק תשובות שייכתבו במקום אחר!**

שאלה 3 (18 נקודות)

נניח שהמחלקה Node שלහלן מ眞מת עצם ביניי.

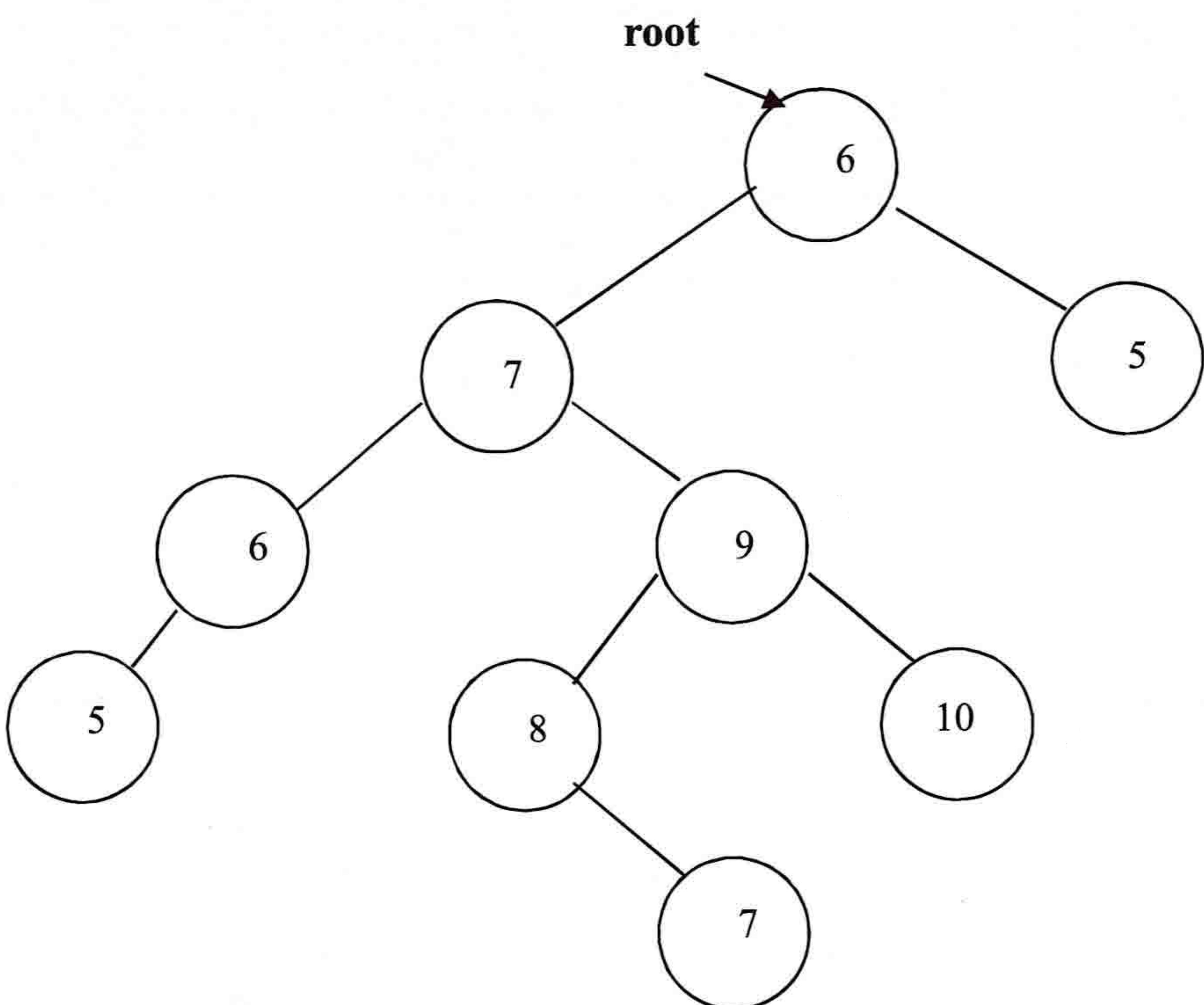
```
public class Node {  
    private int _number;  
    private Node _leftSon, _rightSon;  
  
    public Node (int number) {  
        _number = number;  
        _leftSon = null;  
        _rightSon = null;  
    }  
  
    public int getNumber() {return _number;}  
    public Node getLeftSon() {return _leftSon;}  
    public Node getRightSon() {return _rightSon;}  
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול **בעץ ביניי**.

בין השיטות נתונה השיטה treeSecret הbhא, המתקבלת שורש של עצם ביניי.

```
public static boolean treeSecret (Node ptr)  
{  
    if (ptr == null)  
        return true;  
  
    if (ptr.getLeftSon() == null && ptr.getRightSon() == null)  
        return true;  
  
    if (ptr.getLeftSon() == null)  
        return (Math.abs(ptr.getNumber() - ptr.getRightSon().getNumber()) <= 1) &&  
            treeSecret(ptr.getRightSon());  
  
    if (ptr.getRightSon() == null)  
        return (Math.abs(ptr.getNumber() - ptr.getLeftSon().getNumber()) <= 1) &&  
            treeSecret(ptr.getLeftSon());  
  
    return Math.abs(ptr.getNumber() - ptr.getLeftSon().getNumber())<=1 &&  
        Math.abs(ptr.getNumber() - ptr.getRightSon().getNumber())<=1 &&  
        treeSecret(ptr.getLeftSon()) &&  
        treeSecret(ptr.getRightSon());  
}
```

נתון העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

(3 נק') א. איזה ערך תחזיר השיטה `treeSecret` בעקבות הקריאה `?BinaryTree.treeSecret(root)`

התשובה היא:

False

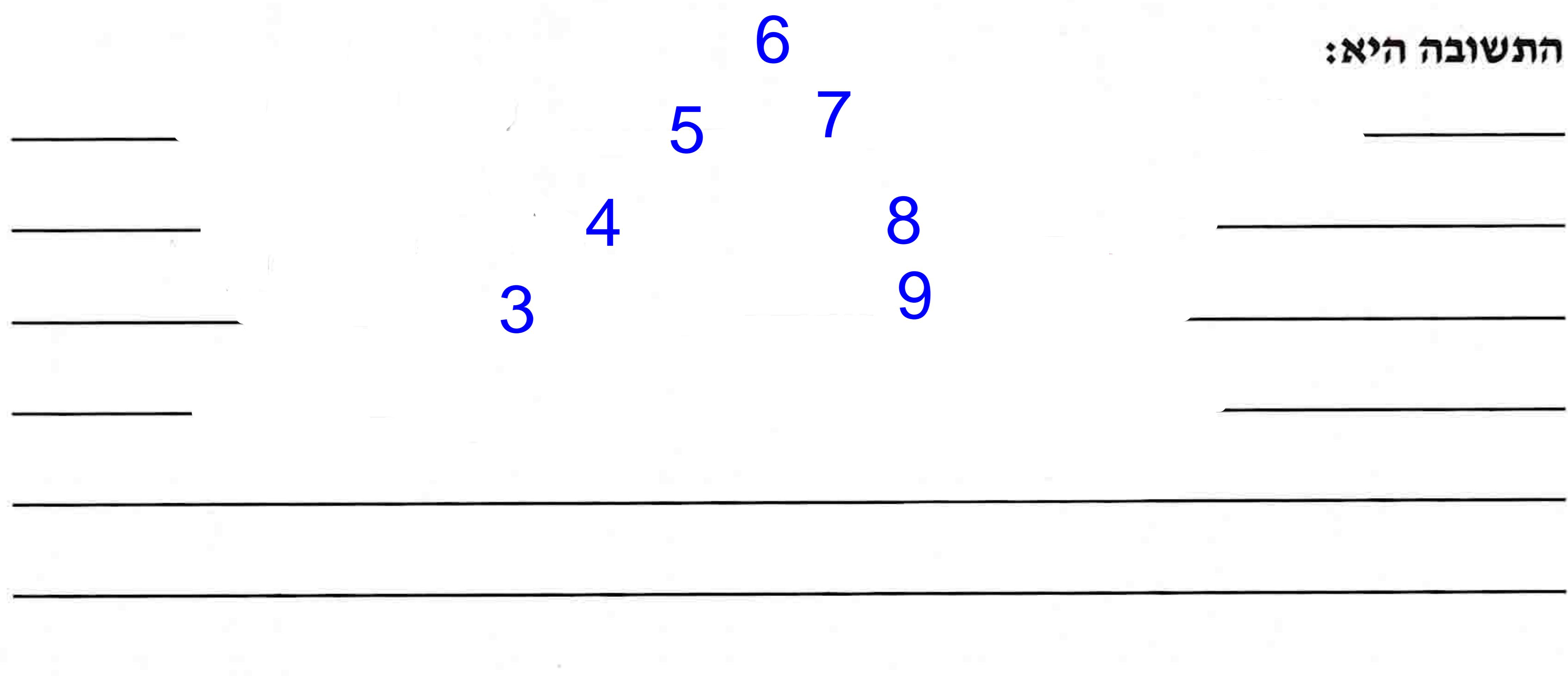
(3 נק') ב. איזה שינוי מינימלי אפשר לעשות על העץ לעיל, כדי שהקריאה לשיטה `treeSecret(root)` תחזיר ערך אחר מזה שהוחזר בסעיף א? **שים לב, השינוי צריך להיות בעץ ולא בשיטה (כגון, הוספת צומת, הורדת צומת, שינוי ערך של צומת ועוד'). מינימלי במבנה של מינימום פעולות על העץ.** כך, אם מורידים צומת, יש לצומת הזה בניים, במספר הפעולות נספרים גם הבנים של הצומת שהורדו.

התשובה היא:

לשנות את הצומת 9 ל- 8 .

לשנות את הצומת 10 ל-9

(5 נק') ג. האם ניתן לצייר עץ חיפוש בינירי (Binary Search Tree) כך שם נריץ את השיטה `treeSecret` עליו, השיטה תחזיר true? אם כן, ציירו עץ כזה בן 5 צמתים לפחות. אם לא, הסבירו מדוע.



(7 נק') ד. מה מבצעת השיטה `treeSecret` באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינירי `root`? שימו לב, עליכם לתת תיאור ממצה של מה עשוה השיטה באופן כללי, ולא תיאור של מה עשוה כל שורה בשיטה, או איך היא מבצעת זאת. כמובן, מה המשמעות של הערך שהשיטה מחזירה? התיחסו למקדי קצה!

התשובה היא:

השיטה בודקת את הפרש בין ערכי הצמתים אם הם קטנים או שווים ל 1 .

במידה ולא , נחזיר False ובמידה שכן , נחזיר True .

המשך הבדיקה בעמוד הבא

שאלה 4 (16 נקודות)

נתון פרויקט שהוגדרו בו המחלקות שלහן. כל אחת בקובץ נפרד, כמוובן.

```

public class A
{
    protected int _x;

    public A()
    {
        _x = 9;
        System.out.println ("A._x = " + _x);
    }

    public A(int x)
    {
        _x = x;
        System.out.println ("A._x = " + _x);
    }
}

//-----

public class B extends A
{
    public B()
    {
        System.out.println ("B._x = " + _x);
    }

    public B(int x)
    {
        super(x);
        System.out.println ("B._x = " + _x);
    }
}

//-----

public class C extends B
{
    public C()
    {
        super();
        System.out.println ("C._x = " + _x);
    }

    public C(int x)
    {
        super(x);
        _x = _x * x;
        System.out.println ("C._x = " + _x);
    }
}

```

```

//-----

public class D extends C
{
    public D()
    {
        super();
        _x++;
        System.out.println ("D._x = " + _x);
    }

    public D(int x)
    {
        super(x);
        System.out.println ("D._x = " + _x);
    }

    public D(int x, int y)
    {
        _x = _x + x + y;
        System.out.println ("D._x = " + _x);
    }
}

//-----
```

```

public class E extends C
{
    public E()
    {
        System.out.println ("E._x = " + _x);
    }
}
```

בפרויקט נמצאת גם המחלקה Tester ובה השיטה main שלහן :

```

public static void main (String[] args)
{
    A a1 = new B();
    B b2 = new B(4);
    C c3 = new B(6);
    C c4 = new C(5);
    C c5 = new D();
    D d6 = new D(2);
    A a7 = new D(3,7);
    D d8 = new E();
}
```

(המשך השאלה בעמוד הבא)

התיחסו לשיטה `main` וכתבו מה קורה לאחר כל פקודה שבשיטה.
 אם הפקודה לא עוברת קומpileציה, כתבו למה.
 אם יש שגיאה בזמן ריצה, כתבו למה.
 אם הכל תקין, כתבו מה יהיה הפלט שיודפס בעקבות הפקודה.
אין קשר בין הפקודות!

1. A a1 = new B();

A._x = 9

B._x = 9

2. B b2 = new B(4);

A._x = 4

B._x = 4

3. C c3 = new B(6);

שגיאת קומPILEציה, C מציין לאב B, לא אפשרי.

4. C c4 = new C(5);

A._x = 5

B._x = 5

C._x = 25

5. C c5 = new D();

A._x = 9

B._x = 9

C._x = 9

D._x = 10

6. D d6 = new D(2);

A._x = 2

B._x = 2

C._x = 4

D._x = 4

7. A a7 = new D(3,7);

A._x = 9

B._x = 9

C._x = 9

D._x = 19

8. D d8 = new E();

שגיאת קומPILEציה אין קשר ירושה בין D ל E.

שאלה 5 (16 נקודות)

נתונה המחלקה IntNode הבאה, המיצגת 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 מקוושרת של מספרים שלמים, המוממשת בעזרת המחלקה IntList שלහלן:

```
public class IntList {  
    private IntNode _head;  
    public IntList( ) {  
        _head = null;  
    }  
    public IntList(IntNode h ) {  
        _head = h;  
    }  
    // כאן יש עוד בנאים וטיפולות... //  
    // המשך המחלקה בעמוד הבא //
```

```

public int what()
{
    int count = 0, res=0, m = -1;
    IntNode temp = _head;
    while (temp !=null)
    {
        if (count == 0)
        {
            m = temp.getValue();
            count = 1;
        }
        else
        {
            if(temp.getValue() == m)
                count++;
            else
                count--;
        }
        res++;
        temp = temp.getNext();
    }
    count = 0;
    temp = _head;
    while (temp !=null)
    {
        if (temp.getValue() == m)
            count++;
        temp = temp.getNext();
    }
    if (count > res/2)
        return m;
    else
        return -9999;
}
. . . // other methods
}

```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגרים מסולסלים.
כך למשל, נסמן $\{ -4, 9, 12, -21 \}$ את הרשימה

סעיף א (2 נקודות)

אם נפעיל את השיטה `what` על הרשימה `{ 4, 2, 1, 3, 2, 5 }` מה יוחזר?

התשובה היא:

9999-

סעיף ב (2 נקודות)

אם נפעיל את השיטה `what` על הרשימה `{ 1, -5, 1, 3, -5, 1, 1, 3, 1 }`, מה יוחזר?

התשובה היא:

1

סעיף ג (2 נקודות)

אם נפעיל את השיטה `what` על הרשימה `{ 2, 2, 1, 3, 2, 5, 3, 2, 2 }` מה יוחזר?

התשובה היא:

2

סעיף ד (2 נקודות)

אם נפעיל את השיטה `what` על הרשימה `{ 1, 0, 1, 0, 1, 0, 1, 0, 0 }` מה יוחזר?

התשובה היא:

9999-

סעיף ה (8 נקודות)

מה מבצעת השיטה `what` באופן כללי כשהיא מופעלת על רשימה מקוורת? הסבירו בקצרה מה השיטה עשוה ולא **כיצד** היא מבצעת זאת.

שים לב, עלייכם לתת תיאור ממצה של מה עשו **השיטה what** באופן כללי, ולא תיאור של מה עשוה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, מה המשמעות של הערך שיזור כתוצאה מהפעלת השיטה על רשימה מקוורת בלבד.

התיחסו למקורי קצת.

התשובה היא:

השיטה בודקת כמה פעמים מופיעה ספרה מסוימת במערך, וסוכמת אותם.

לאחר מכן, השיטה בודקת האם כמות האיבר שחוזר על עצמו גדול ממחצית מהמערך, במידה וכן השיטה תחזיר את אותו מספר, במידה לא השיטה תחזיר - 9999-

ב ה צ ל ח ה !!