

מס' שאלון - 477 26  
באוגוסט 2021

מס' מועד 92

סמסטר 2021ב

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 15 עמודים

מבנה הבחינה:

קראו בעיון את ההנחיות שלהלן:

- \* בבחינה יש חמש 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!
- \* אפשר לתעד בעברית. אין צורך בתיעוד API.

את התשובות לשאלות 3-5 יש לכתוב על גבי השאלון בלבד.  
לא ייבדקו תשובות שייכתבו במקום אחר.

חומר עזר:

חוברות השקפים 1-6, 7-12.

אין להכניס חומר מודפס נוסף או חומר אחר מכל סוג.

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)

נתון מערך חד-ממדי `arr` שערכיו הם מספרים שלמים חיוביים ממש (אין צורך לבדוק זאת).  
בהינתן מספר שלם `num` כלשהו, אנו רוצים לבדוק אם אפשר להגיע למספר `num` על ידי חיבור  
וחיסור חלק מהמספרים במערך `arr`, ובכמה אופנים.  
**לדוגמא,**

אם המערך `arr` הוא להלן והמספר `num` הוא 4,

0	1	2	3
1	3	6	2

אפשר להגיע למספר 4 כך:

- $+3 + 1 = 4$
- $+6 - 3 + 1 = 4$
- $+2 + 3 - 1 = 4$
- $+2 + 6 - 3 - 1 = 4$
- $-2 + 6 = 4$

כלומר, יש כאן 5 ביטויים חשבוניים עם פעולות חיבור וחיסור בלבד, שמערבים את איברי המערך  
`arr` ותוצאותיהם הם המספר  $num = 4$ .

**שימו לב שאנו סופרים פעם אחת כל ביטוי, ולא סופרים את הביטויים שמשתמשים בחוק החילוף.**  
כלומר, אם ספרנו את הביטוי  $+3 + 1$  לא נספור את הביטוי  $+3 + 1$ .

כתבו שיטה סטטית **רקורסיבית** שמקבלת כפרמטרים מערך `arr` של שלמים חיוביים ממש, ומספר  
שלם `num`, ו**מחזירה** את מספר הביטויים החשבוניים שמשתמשים בפעולות חיבור וחיסור על איברי  
המערך (לא בהכרח כולם), ומגיעים לתוצאה `num`.  
כמו כן, השיטה צריכה להדפיס את הביטויים האלו. ההדפסה תכלול את המספר השלם והסימן  
פלוס (+) או מינוס (-) שלפניו, ורווח לאחריו.

ראו את הדוגמא להלן.



דוגמא:

אם נפעיל את השיטה על המערך arr לעיל ועל המספר  $num = 3$  יודפסו הביטויים הבאים:

+2 +1

-2 +6 -1

+3

+6 -3

והשיטה תחזיר את הערך 4.

חתימת השיטה היא:

\\ יש טעות בשאלה לדעתי, צריך להתקבל Num בשיטה.

```
public static int printExpr(int [] arr,int num)
public static int printExpr (int[] arr)
```

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).

שימו לב, שיטה שלא תבצע את ההדפסה של הביטויים החשבוניים, אלא רק תחזיר (באופן נכון) את מספרם, תקבל לכל היותר 15 נקודות.

אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות! אל תשכחו לתעד את מה שכתבתם!



```
public static int printExpr(int [] arr,int num) {  
    return printExpr(arr,num,0,0,"");  
}  
  
private static int printExpr(int [] arr,int num,int i,int sum,String str) {  
  
    if(sum == num) {  
        System.out.println(str);  
        return 1;  
    }  
    if(i > arr.length-1) {  
        return 0;  
    }  
  
    int way1 = printExpr(arr,num,i+1,sum,str);  
    int way2 = printExpr(arr,num,i+1,sum+arr[i],str + "+" + arr[i]);  
    int way3 = printExpr(arr,num,i+1,sum-arr[i],str + "-" + arr[i]);  
    return way1+way2+way3;  
}
```



שאלה 2 (25 נקודות)

נתון מערך חד-ממדי שמכיל רק אפסים (0) ואחדים (1).

כתבו שיטה סטטית longestSequence המקבלת כפרמטרים מערך a כזה ומספר שלם אי-שלילי k. השיטה צריכה להחזיר את האורך של התת-סדרה הרציפה הארוכה ביותר של המערך a, שכוללת לכל היותר k אפסים.

לדוגמא:

אם נתון המערך a הבא:

0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	1	0	1	1	1	1	0	0

- עבור  $k = 0$   
השיטה תחזיר 4 הסדרה באורך 4 המתחילה ממקום 6 היא הארוכה ביותר עם  $k=0$  אפסים
- עבור  $k = 1$   
השיטה תחזיר 7 (הסדרה באורך 7 המתחילה ממקום 3 היא הארוכה ביותר עם  $k=1$  אפסים)
- עבור  $k = 2$   
השיטה תחזיר 10 (הסדרה באורך 10 המתחילה ממקום 0 מכילה בדיוק  $k=2$  אפסים)
- עבור  $k = 3$   
השיטה תחזיר 11 (הסדרה באורך 11 המתחילה ממקום 0 מכילה מספר אפסים מקסימלי האפשרי 3 אפסים)
- עבור  $k = 4$  וכל המספרים k שגדולים מ-4  
השיטה תחזיר 12 (הסדרה באורך 12 המתחילה ממקום 0 מכילה מספר אפסים מקסימלי האפשרי 4 אפסים, והיא בעצם המערך כולו)

חתימת השיטה:

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

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.  
מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם?  
הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!



```
public static int longestSequence(int [] a, int k) {  
    int left = 0;  
    int counter = 0;  
    int sum = k;  
    while (left + counter < a.length)  
    {  
        sum += (a[left + counter] - 1);  
        // System.out.println(a[max + start] - 1);  
        if (sum >= 0)  
            counter++;  
        else  
        {  
            sum -= (a[left] - 1); // so it will ignore the current place  
            left++;  
        }  
    }  
    return counter;  
}
```



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

שאלה 3 (18 נקודות)

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

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

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

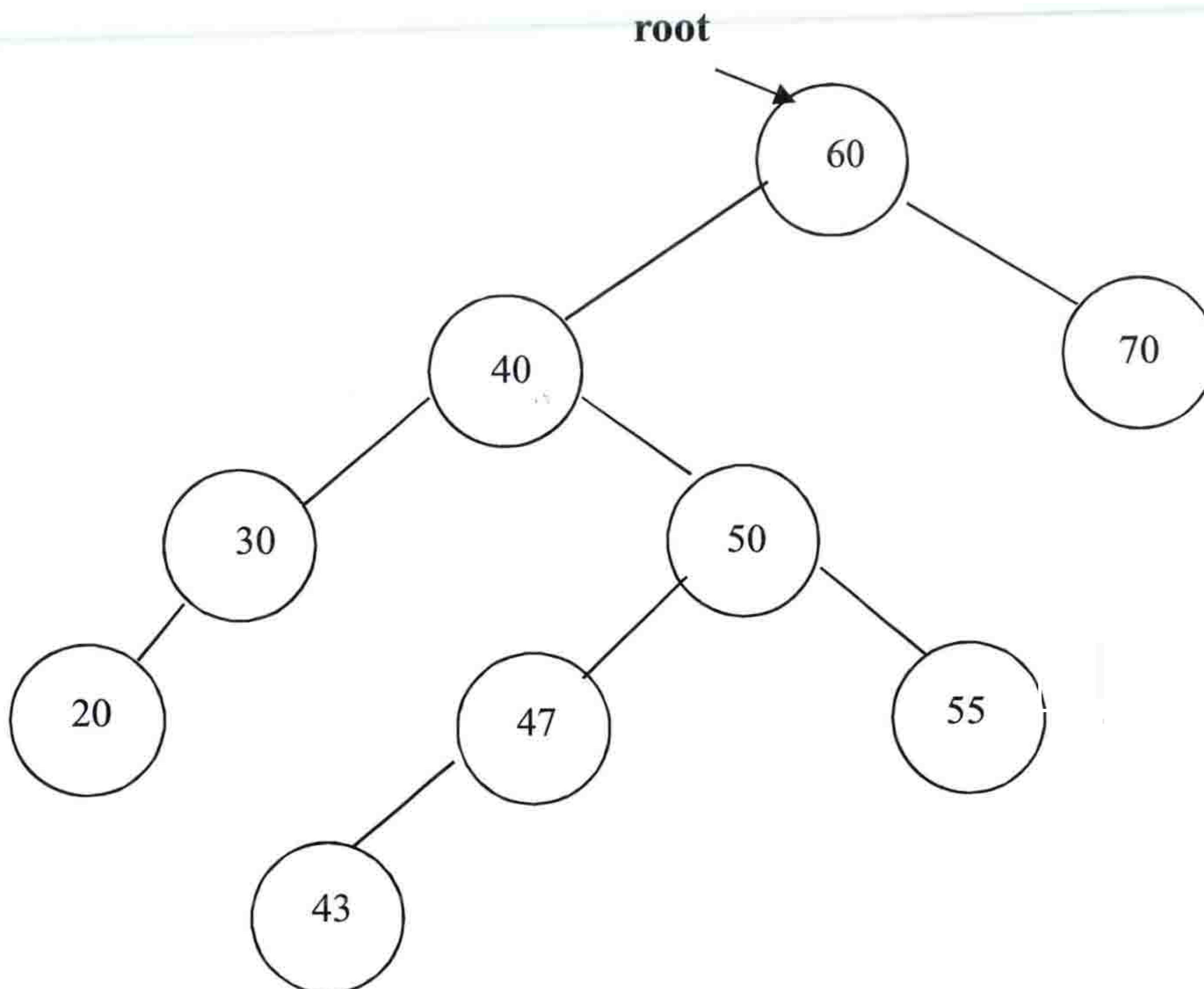
    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }
}
```

המחלקה BinarySearchTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ חיפוש בינרי.  
נתונה השיטה find הבאה, המקבלת שורש של עץ חיפוש בינרי ומספר שלם כלשהו.

```
public static int find(Node root, int num)
{
    if (root == null)
        return -1;
    if (root.getNumber() == num)
        return num;
    else if (root.getNumber() < num) {
        int k = find(root.getRightSon(), num);
        if (k == -1)
            return root.getNumber();
        else
            return k;
    }
    else if (root.getNumber() > num)
        return find(root.getLeftSon(), num);
    return -1;
}
```



נתון עץ החיפוש הבינרי הבא, ששורשו הוא root.  
אנחנו מניחים שכל הערכים שנמצאים בעץ הם חיוביים!



ענו על הסעיפים הבאים:

2 נק' (א) איזה ערך תחזיר השיטה **find** בעקבות הקריאה `BinarySearchTree.find(root, 43)`?  
התשובה

43

2 נק' (ב) איזה ערך תחזיר השיטה **find** בעקבות הקריאה `BinarySearchTree.find(root, 59)`?  
התשובה היא:

55

2 נק' (ג) איזה ערך תחזיר השיטה **find** בעקבות הקריאה `BinarySearchTree.find(root, 15)`?  
התשובה היא:

1-



3 נק') (ד) מה מבצעת השיטה **find** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ חיפוש בינרי root ומספר שלם num 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 שהשיטה מחזירה? התייחסו למקרי קצה!

התשובה היא:

העץ בכיוון צד שמאל יחזיר 1 אם הוא לא ימצא את הערך בתוך העץ.  
העץ בכיוון צד ימין יחזיר את הצומת עצמה אם הגיעו לצומת ימנית ריקה ועדין לא נמצא הערך.  
במידה ונמצא הערך בתוך העץ נחזיר את אותו ערך.

9 נק') (ה) הסעיף הזה לא קשור לסעיפים הקודמים!

עליכם לצייר עץ חיפוש בינרי (Binary Search Tree) ששורשו root כך שאם נדפיס אותו בסדר סופי (post order) יתקבל הפלט הבא (משמאל לימין):

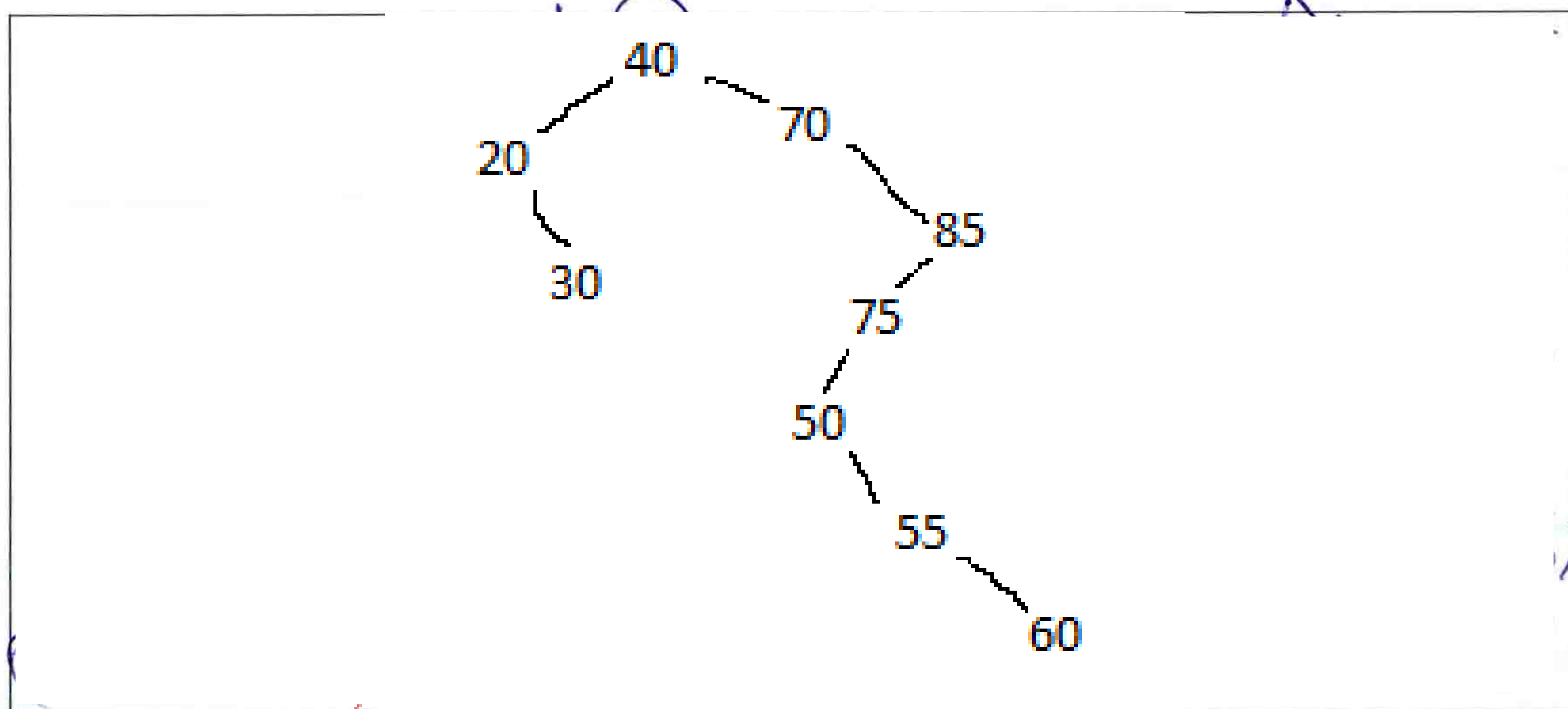
30 20 60 55 50 75 85 70 40

אם לא קיים עץ חיפוש בינרי כזה, עליכם להסביר מדוע.

להזכירכם, השיטה postOrder המדפיסה עץ בינרי בסדר סופי ממומשת כך:

```
public static void printPostOrder (Node root)
{
    if (root != null)
    {
        printPostOrder(root.getLeftSon());
        printPostOrder(root.getRightSon());
        System.out.print (root.getNumber() + " ");
    }
}
```

התשובה היא (ציירו את העץ במסגרת להלן):



0  
(3)



נתון פרויקט שהוגדרו בו המחלקות A, B, C ו-Driver שלהלן. כל אחת בקובץ נפרד, כמובן.

```
public abstract class A
{
    protected int _x;

    public A() {
        _x = 1;
    }

    public abstract int foo(int x);

    public void foo(A a) {
        _x = a._x;
    }
}
```

```
public class B extends A
{
    public B() {
        super();
    }

    public B(int val) {
        _x = val;
    }

    public B(A val) {
        _x = foo(val._x);
    }

    public int foo(int x) {
        return _x + x;
    }

    public void foo(B b) {
        _x = _x * b._x;
    }
}
```



```
public class C extends B {

    public C() {
        super();
    }

    public C(int val) {
        super(val);
    }

    public int foo(int x) {
        return _x - x;
    }

    public void foo(A a) {
        if (a instanceof C)
            _x = _x - a._x;
        else
            super.foo(a);
    }

    public void foo(B b) {
        super.foo(b);
        _x++;
    }

    public void foo(C c) {
        _x = c._x - 1;
    }

}
```

במחלקה Driver נמצאת השיטה main שלהלן:

```
public static void main(String[] args)
{
    A a;
    B b;
    C c;

    - כאן יופיעו הפקודות שבסעיפים הבאים -
}
```

עבור כל סעיף להלן, הקיפו בעיגול את התשובה המתאימה ביותר. עליכם להחליט האם קטע הקוד גורם לשגיאת קומפילציה, שגיאת ריצה, או תקין. במידה והקוד תקין השלימו את התשובה וכתבו מה יהיה ערכו של `_x`.

אין קשר בין הסעיפים!



```
/*1*/ a = new A();
      a.foo(2);
```

שגיאת קומפילציה אי אפשר להשתמש בA . הוא אבסטרקטי.

קטע הקוד גורם לשגיאת קומפילציה והיא \_  
קטע הקוד גורם לשגיאת ריצה והיא  
קטע הקוד תקין וערך התכונה x\_ באובייקט a

```
/*2*/ a = new B();
      b = new C();
      b.foo(a);
```

—  
—————

1

—————

קטע הקוד גורם לשגיאת קומפילציה והיא \_  
קטע הקוד גורם לשגיאת ריצה והיא  
קטע הקוד תקין וערך התכונה x\_ באובייקט a

```
/*3*/ a = new C();
      b = new C();
      ((C)b).foo(a);
```

1

קטע הקוד גורם לשגיאת קומפילציה והיא \_  
קטע הקוד גורם לשגיאת ריצה והיא  
קטע הקוד תקין וערך התכונה x\_ באובייקט a

```
/*4*/ C c1 = new C();
      b = new C();
      b.foo(c1);
```

2

קטע הקוד גורם לשגיאת קומפילציה והיא \_  
קטע הקוד גורם לשגיאת ריצה והיא  
קטע הקוד תקין וערך התכונה x\_ באובייקט b

```
/*5*/ c = new C();
      c.foo(c);
```

0

טע הקוד גורם לשגיאת קומפילציה והיא \_  
טע הקוד גורם לשגיאת ריצה והיא \_  
טע הקוד תקין וערך התכונה x\_ באובייקט c הוא :



```
/*6*/ C c1 = new C(2);
```

```
a = new C(3);
```

```
c1.foo((B)a);
```

ע הקוד גורם לשגיאת קומפילציה והיא \_\_\_\_\_

ע הקוד גורם לשגיאת ריצה והיא \_\_\_\_\_

ע הקוד תקין וערך התכונה `_x` באובייקט `a` הוא: 3

```
/*7*/ b = new B(2);
```

```
Object o = b;
```

```
o.foo(4);
```

הקוד גורם לשגיאת קומפילציה והיא \_\_\_\_\_

הקוד גורם לשגיאת ריצה והיא \_\_\_\_\_

הקוד תקין וערך התכונה `_x` באובייקט `a` הוא: 3

```
/*8*/ C c1 = new C(2);
```

```
a = new C(3);
```

```
c1.foo(a);
```

זקוד גורם לשגיאת קומפילציה והיא \_\_\_\_\_

זקוד גורם לשגיאת ריצה והיא \_\_\_\_\_

זקוד תקין וערך התכונה `_x` באובייקט `a` הוא: 3

3

(4)



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 what1 (IntList list)
    {
        IntNode h1 = _head;
        IntNode h2 = list._head;
        while ((h1 != null) && (h2 != null))
        {
            if (h1.getValue() != h2.getValue())
                return false;
            h1 = h1.getNext();
            h2 = h2.getNext();
        }
        return true;
    }
}
```

לדעתי בחלק הזה יש שגיאה - <  
 הקוד היה צריך להיות כזה:  
 public intList(IntNode node){  
 \_head = node;  
 }



```

public boolean what2 (IntList list)
{
    IntNode h1 = _head;

    while (h1 != null)
    {
        boolean found = false;
        IntNode h2 = list._head;
        while ((h2 != null) && (!found))
        {
            if (h1.getValue() == h2.getValue())
                found = true;
            h2 = h2.getNext();
        }
        if (!found)
            return false;
        h1 = h1.getNext();
    }
    return true;
} //end of class IntList

```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.  
 כך לדוגמא, נסמן { -4 , 9 , 12 , -21 } את הרשימה  $-4 \rightarrow 9 \rightarrow 12 \rightarrow -21 \rightarrow \text{null}$

ענו על ששת הסעיפים להלן:

\\ אני חושב שיש טעות בשאלה הזו כי `head_` מאותחל ל `Null`.  
 בכל הסעיפים נתונה הרשימה `testList = {1, 2, 3, 4, 5}`.

סעיף א (2 נקודות)

תנו דוגמא של רשימה בת 4 איברים לפחות בשם `list` כך שהקריאה `testList.what1(list)` תחזיר `true` והקריאה `testList.what2 (list)` תחזיר `true`.  
 אם לא יכולה להיות רשימה כזו, כתבו זאת והסבירו מדוע.  
 התשובה היא:

1 2 3 4 5

---



---



---



סעיף ב (2 נקודות)

תנו דוגמא של רשימה בת 4 איברים לפחות בשם list כך שהקריאה testList.what1(list) תחזיר true והקריאה testList.what2(list) תחזיר false.  
אם לא יכולה להיות רשימה כזו, כתבו זאת והסבירו מדוע.  
התשובה היא:

אם יש טעות בשאלה -  
נשלח את הרשימה 1 2 3 4  
אם אין טעות בשאלה -  
לא ניתן, h1, h2 יאותחלו תמיד כ Null ולכן תמיד יחזור true משתי השיטות.

סעיף ג (2 נקודות)

תנו דוגמא של רשימה בת 4 איברים לפחות בשם list כך שהקריאה testList.what1(list) תחזיר false והקריאה testList.what2(list) תחזיר true.  
אם לא יכולה להיות רשימה כזו, כתבו זאת והסבירו מדוע.  
התשובה היא:

אם יש טעות בשאלה -  
נשלח את הרשימה 1 2 4 3 5  
אם אין טעות בשאלה -  
לא ניתן h1, h2 יאותחלו תמיד כ Null ולכן תמיד יחזור true משתי השיטות

סעיף ד (2 נקודות)

תנו דוגמא של רשימה בת 4 איברים לפחות בשם list כך שהקריאה testList.what1(list) תחזיר false והקריאה testList.what2(list) תחזיר false.  
אם לא יכולה להיות רשימה כזו, כתבו זאת והסבירו מדוע.  
התשובה היא:

אם יש טעות בשאלה -  
נשלח את הרשימה 1 2 4 3  
אם אין טעות בשאלה -  
לא ניתן h1, h2 יאותחלו תמיד כ Null ולכן תמיד יחזור true משתי השיטות



**סעיף ה (4 נקודות)**

מה מבצעת השיטה what1 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 what1, כשהיא מופעלת על רשימה כלשהי list1 ומקבלת כפרמטר רשימה אחרת list2.

**התשובה היא:**

השיטה משווה בין רשימה נוכחית לרשימה שמתקבלת כפרמטר, במידה וערכיהם שונים יוחזר False אחרת יוחזר True.

---

---

---

---

---

---

---

---

**סעיף ו (4 נקודות)**

מה מבצעת השיטה what2 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 what2, כשהיא מופעלת על רשימה כלשהי list1 ומקבלת כפרמטר רשימה אחרת list2.

**התשובה היא:**

השיטה בודקת האם כל איברי הרשימה הנוכחית קיימים בתוך הרשימה המתקבלת כפרמטר

---

---

---

---

---

---

---

---

**בהצלחה!**