

20441 / 4

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

שאלה 1 (25 נקודות)

אדם צריך להגיע ממקום אחד לאחר. בדרכו הוא יכול לעבור בכל מיני תחנות. בכל תחנה יש מספר מטילי זהב במשקלים שונים שהוא יכול לקחת. כמובן שהוא רוצה לקחת את המשקל הגדול ביותר של מטילי הזהב האפשרי. העניין הוא שהוא לא יכול לעבור מתחנה לתחנה סתם כך, אלא לפי תנאים מסוימים שיפורטו להלן.

נניח שהמשקלים של מטילי הזהב שיש בתחנות מאוחסנים במערך חד-ממדי. בכל תא במערך קיימים מטילי זהב במשקל מסוים, שהוא שלם חיובי ממש (לא אפס). **האדם נמצא בתא הראשון במערך והוא מעוניין להגיע אל התא האחרון.**

בכל צעד (ספירת הצעדים מתחילה מ-1) הוא יכול לקחת קפיצה שגודלה הוא הסכום של הערך הנמצא במערך בנקודה בו הוא נמצא ומספרו של הצעד.

מותר לבקר בכל תא (כולל תא מספר 0) במערך לכל היותר פעם אחת. הקפיצה יכולה להיות ימינה או שמאלה, אך אסור לה להוביל אל מחוץ לגבולות המערך.

לדוגמא, אם המערך המכיל את משקלי מטילי הזהב הוא זה:

0	1	2	3	4	5	6	7	8
2	1	2	3	1	4	1	3	1

אז המסלול היחיד האפשרי הוא:

מתחילים בתא 0. לוקחים את המטיל שמשקלו 2. בצעד הראשון אפשר ללכת רק ימינה, והקפיצה היא בגודל 3 = הסכום של מספר הצעד (1) + הערך בתא (2). קופצים לתא 3. בתא 3 לוקחים את המטיל שמשקלו 3. הקפיצה הבאה האפשרית היא בגודל 5 = הסכום של מספר הצעד (2) + הערך בתא (3). לכן אי אפשר לקפוץ שמאלה, אלא רק ימינה ומגיעים מתא 3 לתא 8 שהוא התא האחרון במערך. לוקחים את המטיל האחרון שמשקלו 1. סה"כ המשקל של מטילי הזהב שנלקחו הוא 6 (2 בתא 0, 3 בתא 3, 1 בתא 8, $6 = 2+3+1$).

בעמוד הבא תמצאו דוגמא נוספת.

כתבו שיטה סטטית רקורסיבית המקבלת כפרמטר מערך חד-ממדי המכיל מספרים שלמים חיוביים ממש, ומחזירה את המשקל הגדול ביותר של מטילי זהב שאפשר לקחת אם הולכים במסלולים האפשריים מהתא הראשון לתא האחרון במערך. כמו כן, השיטה צריכה להדפיס את המסלולים האפשריים, ואת הסכום של המטילים בכל אחד מהם.

אם אין שום מסלול אפשרי שמוביל מהתא הראשון במערך אל התא האחרון, השיטה לא צריכה להדפיס כלום, ועליה להחזיר את הערך `Integer.MAX_VALUE`. (כך גם, אם יש מסלול שמגיע למבוי סתום כי אי אפשר לעשות ממנו שום צעד חוקי, השיטה צריכה פשוט להתעלם ממנו ולא להדפיס אותו. ההדפסה צריכה להיות רק כאשר מגיעים אל היעד).

חתימת השיטה היא:

```
public static int heaviestPath (int [] arr)
```

דוגמא נוספת:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
arr =	9	4	7	7	4	4	1	3	1	2	1	3	2	1	9	2

במערך זה יש בדיוק ארבעה מסלולים אפשריים. לכן אם השיטה שלכם תקבל את המערך הזה, הפלט צריך להראות בדיוק כך :

1. 0 -- 10 -- 13 -- 9 -- 15 sum = 15
2. 0 -- 10 -- 13 -- 9 -- 3 -- 15 sum = 22
3. 0 -- 10 -- 7 -- 13 -- 8 -- 2 -- 15 sum = 24
4. 0 -- 10 -- 7 -- 1 -- 9 -- 2 -- 15 sum = 28

המספרים הכתובים הם מספרי האינדקסים של התאים שמשתתפים במסלול, משמאל לימין. בסוף כל מסלול כתוב הסכום של משקלי המטילים שנלקחו במסלול זה. שימו לב שהסכום הוא של הערכים שבתאים ולא של האינדקסים.

לאחר הדפסת ארבעת המסלולים כמתואר לעיל, השיטה צריכה להחזיר את הערך 28, שהוא המירבי בין סכומי משקלי המטילים בכל המסלולים האפשריים.

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

שימו לב:

- אם השיטה שתכתבו רק תחזיר את המשקל המקסימלי, אבל לא תדפיס את המסלולים, התשובה תקבל לכל היותר 17 נקודות.
- מותר לשנות את המערך במהלך השיטה, אבל המערך צריך לחזור לקדמותו לאחר ביצוע השיטה.
- מותר להשתמש בשיטות `Math.max` ו-`Math.min` מהמחלקה `Math` וכן בקבועים `Integer.MAX_VALUE`, `Integer.MIN_VALUE` מהמחלקה `Integer`.
- מותר להשתמש בהעמסת-יתר (`Overloading`)
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

שאלה 2 (25 נקודות)

נתון 2 מערכים חד-ממדיים a ו- b , שניהם באורך זהה, המכילים מספרים שלמים.

סעיף א (20 נקודות):

כתבו שיטה סטטית המקבלת כפרמטרים את שני המערכים a ו- b ומוצאת זוג איברים, אחד מהמערך a והשני מהמערך b כך שההפרש (בערך מוחלט) בין שני האיברים האלו הוא **הקטן ביותר** מבין כל ההפרשים של הזוגות האפשריים. השיטה צריכה להחזיר את ההפרש ולהדפיס (בתוך השיטה) את זוג האיברים.

חתימת השיטה היא:

```
public static int smallestDiffPair(int[] a , int[] b)
```

סעיף ב (5 נקודות):

כתבו שיטה סטטית המקבלת כפרמטרים את שני המערכים a ו- b ומוצאת זוג איברים, אחד מהמערך a והשני מהמערך b כך שההפרש (בערך מוחלט) בין שני האיברים האלו הוא **הגדול ביותר** מבין כל ההפרשים של הזוגות האפשריים. השיטה צריכה להחזיר את ההפרש ולהדפיס (בתוך השיטה) את זוג האיברים.

חתימת השיטה היא:

```
public static int biggestDiffPair(int[] a , int[] b)
```

לדוגמא, אם המערכים הם:

	0	1	2	3	4		0	1	2
a =	1	5	-2	6	2	b =	8	3	7

השיטה `smallestDiffPair` תחזיר את הערך 1 ותדפיס את האיברים 2---3 או את 6---7.

השיטה `biggestDiffPair` תחזיר את הערך 10 ותדפיס את האיברים 2---8.

שימו לב:

שתי השיטות שתכתבו צריכות להיות יעילות 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. בשאלה זו, סיבוכיות המקום לא חייבת להיות קבועה אבל היא צריכה להיות מינימלית ככל הניתן.

ציינו מהי סיבוכיות זמן הריצה ומהי סיבוכיות המקום של כל שיטה שכתבתם. הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!

חלק ב - את התשובות לשאלות 3, 4 ו- 5 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!

שאלה 3 (16 נקודות)

נניח שהמחלקה Node שלהלן מממשת צומת בעץ 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; }

    public void setNumber(int number) { _number = number; }
    public void setLeftSon(Node node) { _leftSon = node; }
    public void setRightSon(Node node) { _rightSon = node; }
}
```

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

בין השיטות נתונות השיטות max, f, g ו- what הבאות:

```
public static int max (int a, int b, int c)
{
    return Math.max (a, Math.max(b,c));
}

public static int f (Node t) - הערך גבוה ביותר
{
    if (t == null)
        return 0;
    return t.getNumber() + Math.max (f(t.getLeftSon()), f(t.getRightSon()));
}
```

```

public static int g(Node root)
{
    return root.getNumber() (+)
        f(root.getLeftSon()) + f(root.getRightSon());
}

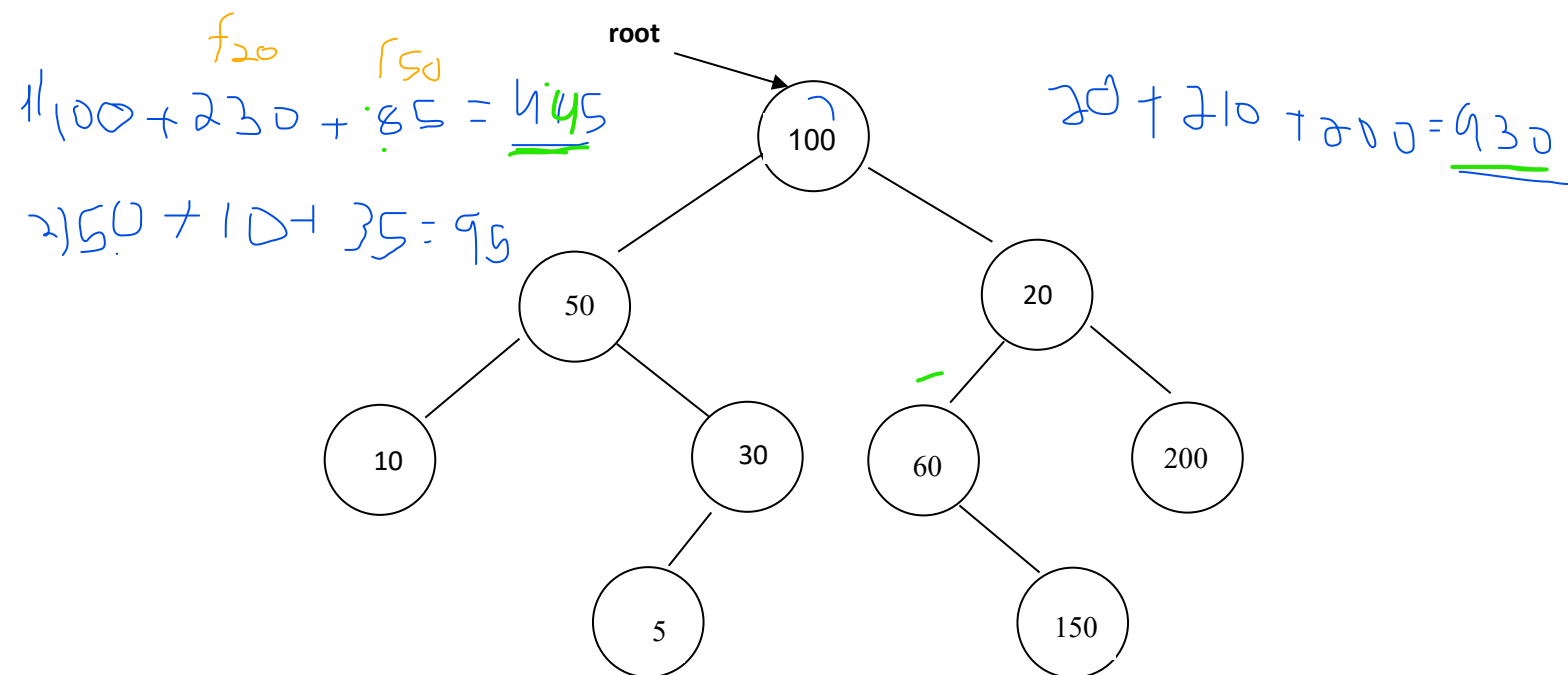
public static int what (Node root)
{
    if (root == null)
        return 0;
    if (root.getLeftSon() == null && root.getRightSon() == null)
        return root.getNumber();
    return max(g(root),
        what (root.getLeftSon()),
        what (root.getRightSon()));
}

```

מספר +
 כי קטנים
 של כל הנוסחה
 - + Max -

השיטות האלו פועלות על עץ בינרי שכל ערכיו הם מספרים חיוביים ממש.

נתון העץ הבינרי הבא, ששורשו הוא root:



ענו על הסעיפים בעמודים הבאים.

סעיף א (2 נקודות):

מה תחזיר השיטה f בעקבות הקריאה $\text{BinaryTree.f}(\text{root})$?

התשובה היא:

330

סעיף ב (4 נקודות):

מה מבצעת השיטה f באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי root ? 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, לכתוב בקצרה מה משמעות הערך שהשיטה מחזירה. התייחסו למקרי קצה.

התשובה היא:

סעיף ג (3 נקודות):

מה תחזיר השיטה what בעקבות הקריאה $\text{BinaryTree.what}(\text{root})$?

התשובה היא:

430

סעיף ד (3 נקודות):

אם נחליף את הצומת שערכו 30 לצומת שערכו 70, מה תחזיר השיטה what בעקבות הקריאה $\text{BinaryTree.what}(\text{root})$?

התשובה היא:

445

סעיף ה (4 נקודות):

מה מבצעת השיטה what באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי root ? 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, לכתוב בקצרה מה משמעות הערך שהשיטה מחזירה. התייחסו למקרי קצה.

התשובה היא:

שאלה 4 (18 נקודות)

נתונה המחלקה `IntNode` הבאה, המייצגת איבר ברשימה מקושרת חד-סטרית המכילה מספרים שלמים:

```
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 חד-סטרית, הממומשת בעזרת המחלקה `IntList` שמכילה את השיטה `secret` הכתובה להלן. אפשר להניח שיש שיטות נוספות במחלקה.

```
public class IntList
{
    private IntNode _head;

    public IntList()           { _head = null; }
    public IntList(IntNode node) { _head = node; }
```

--- המשך המחלקה בעמוד הבא ---

```

public void secret (IntList other)
{
    boolean flag = true;
    int count1 = 0;
    int count2 = 0;
    IntNode search1 = _head;

    while (search1 != null && flag)
    {
        IntNode search2 = other._head;
        while (search2 != null && flag)
        {
            search1 = search1.getNext();
            search2 = search2.getNext();
            count2++;
            if (search1 == null && search2 != null)
            {
                System.out.println("count1 = " + count1 +
                                   " count2 = " + count2);
                flag = false;
            }
            count1++;
            count2 = 0;
        }
        if (flag)
            System.out.println("count1 = " + count1 +
                               " count2 = " + count2);
    }
} //end of class IntList

```

בשאלות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים. כאשר המספר **השמאלי** מסמן את ראש הרשימה.

סעיף א (2 נקודות)

אם נפעיל את השיטה **secret** על הרשימה {1, 2, 3, 4, 5, 6, 7, 8}, כאשר הפרמטר **other** הוא הרשימה {3, 2, 1}. מה יודפס על הפלט? אם, לדעתכם, תהיה שגיאת ריצה, הסבירו מהי בדיוק.

התשובה היא:

$C_1 = 2$ $C_2 = 8$

סעיף ב (2 נקודות)

אם נתונה הרשימה $list2 = \{1, 2, 3, 4\}$, כתבו דוגמא לרשימה $list1$ כך שאם נפעיל את השיטה כך: $list1.secret(list2)$ יודפס הפלט:

$count1 = 2$ $count2 = 2$

אם, לדעתכם, לא יכולה להיות רשימה כזו, נמקו מדוע.

התשובה היא:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

סעיף ג (3 נקודות)

אם נתונה הרשימה $list2 = \{0, 1, 0\}$, כתבו דוגמא לרשימה $list1$ כך שאם נפעיל את השיטה כך: $list1.secret(list2)$ יודפס הפלט:

$count1 = 2$ $count2 = 3$

אם, לדעתכם, לא יכולה להיות רשימה כזו, נמקו מדוע.

התשובה היא:

—

סעיף ד (3 נקודות)

תנו דוגמאות לשתי רשימות **שונות** $list1$ ו- $list2$, כך שכשנפעיל את השיטה **secret** כך: $list1.secret(list2)$ הפלט שיודפס יהיה:

$count1 = 3$ $count2 = 0$

וכשנפעיל את השיטה **secret** כך:

$list2.secret(list1)$ הפלט שיודפס יהיה:

$count1 = 0$ $count2 = 3$

אם, לדעתכם, לא יכולות להיות כאלו רשימות, נמקו מדוע.

התשובה היא:

סעיף ה (3 נקודות)

תנו דוגמאות לשתי רשימות **שונות** list1 ו- list2 כך שאם נפעיל את השיטה secret פעם אחת כך: list1.secret(list2) ופעם שניה כך: list2.secret(list1), הפלט שיודפס בשתי הפעמים יהיה זהה.

אם, לדעתכם, לא יכולות להיות כאלו רשימות, נמקו מדוע.

התשובה היא:

סעיף ו (5 נקודות)

מה מבצעת השיטה **secret** באופן כללי, כאשר היא פועלת על רשימה list1 כלשהי, ומקבל כפרמטר רשימה list2 כלשהי?

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.

חשוב מאד שתסבירו בצורה מדויקת מה המשמעות של הערכים המודפסים count1 ו- count2.

התייחסו למקרי קצה.

התשובה היא:

שאלה 5 (16 נקודות)

נגדיר מערך דו-ממדי המלא במספרים שלמים, כמערך **ממוין-יורד** אם כל שורה במערך ממוינת בסדר יורד ממש (כלומר אין בה מספרים זהים), וכל עמודה במערך ממוינת בסדר יורד ממש.

לדוגמא, המערך m להלן הוא ממוין-יורד:

	0	1	2	3	4	5	6	7
0	82	60	20	10	5	4	-8	-15
1	20	18	2	1	0	-10	-20	-30
2	18	15	0	-30	-50	-55	-62	-70
3	-22	-38	-39	-40	-55	-60	-70	-90

במחלקה Q5 נתונה השיטה הסטטית הבאה, הפועלת על מערך דו-ממדי **ממוין-יורד**:

```
public class Q5
{
    public static int find (int [][] mat)
    {
        int res = 0;
        int row = mat.length-1, col = 0;
        while (row>=0 && col<mat[0].length)
        {
            if (mat[row][col] < 0)
            {
                res += mat[0].length-col;
                row--;
            }
            else
                col++;
        }
        return res;
    }
}
```

סעיף א (3 נקודות):

מה תחזיר השיטה find בעקבות הקריאה find(m), בהינתן לה כפרמטר המערך m שלעיל?
התשובה היא:

18

סעיף ב (6 נקודות)

נניח שנתון מערך דו-ממדי m, שמלא במספרים שלמים ו**ממוין-יורד**. מה מבצעת השיטה find אם נפעיל אותה עם הפרמטר שהוא המערך m? 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. **הסבירו מה המשמעות של הערך המוחזר מהשיטה find**. התייחסו למקרי קצה.

התשובה היא:

נגדיר מערך דו-ממדי המלא במספרים שלמים, כמערך **ממוין-עולה** אם כל שורה במערך ממוינת בסדר עולה ממש, וכל עמודה במערך ממוינת בסדר עולה ממש.

למשל, המערך m לעיל יהיה עכשיו המערך m1 הזה:

	0	1	2	3	4	5	6	7
0	-90	-70	-60	-55	-40	-39	-38	-22
1	-70	-62	-55	-50	-30	0	15	18
2	-30	-20	-10	0	1	2	18	20
3	-15	-8	4	5	10	20	60	82

סעיף ג (7 נקודות)

נניח שנתון מערך דו-ממדי $m1$, שמלא במספרים **שלמים וממוין-עולה**. אנחנו רוצים לכתוב שיטה `find1` שתקבל את המערך $m1$ כפרמטר ותבצע בדיוק את **מה** שעושה השיטה `find`. כלומר, המשמעות של הערך המוחזר מ- `find` תהיה אותה משמעות של הערך המוחזר מ- `find1`.

מהם השינויים המינימליים שצריך לעשות לשם כך בשיטה `find`?

כתבו את תשובתכם בעמוד הבא בשאלון (ורק שם!).

לא בהכרח כל השורות הריקות יתמלאו!

שימו לב שכדאי מאד להיצמד לשיטה `find` ולשנות אותה, ולא לכתוב שיטה מחדש.

```
public static int find1 (int [][] mat)
{
    int res = 0;

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    return res;
}
```

ב ה צ ל ח ה!