

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 13 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 של ד"ר אמיר גורן ותמר וילנר.
יחידות 1-6, 7-12. 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

סעיף א (10 נקודות)

נגדיר: מערך חד-ממדי vec באורך n נקרא **מאוזן** סביב אינדקס p כאשר $0 \leq p < n$ אם מתקיים שסכום האיברים במערך מהאיבר 0 עד לאיבר $p-1$ שווה לסכום האיברים במערך מהאיבר p ועד לאיבר $n-1$.

בנוסחה המתמטית זה נכתב כך: $\sum_{i=0}^{p-1} vec[i] = \sum_{i=p}^{n-1} vec[i]$

לדוגמא, המערך $\{5, 6, 1, 2, 8\}$

0	1	2	3	4
5	6	1	2	8

מאוזן סביב האינדקס 2, שכן: $5 + 6 = 1 + 2 + 8$

כתבו שיטה סטטית **רקורסיבית** *where* המקבלת מערך חד ממדי vec מלא במספרים שלמים, ומחזירה אינדקס שסביבו vec מאוזן. אם אין אינדקס כזה, השיטה מחזירה -1.
חתימת השיטה *where* היא:

```
public int where (int [] vec)
```

לנוחותכם, כדאי לכם להגדיר שיטה פרטית

```
private int where (int [] vec, int left, int p, int right)
```

המקבלת ארבעה פרמטרים:

- vec – המערך;
- $left$ - האינדקס של התא השמאלי ביותר ב- vec ;
- p - אינדקס של תא ב- vec שסביבו vec עשוי להיות מאוזן;
- $right$ - האינדקס של התא הימני ביותר ב- vec .

הניחו שיש לרשותכם שיטה מהצורה `int sum (int [] vec, int lo, int hi)` שמחזירה את הסכום מהאינדקס lo עד לאינדקס hi במערך. (כלומר את $\sum_{i=lo}^{hi} vec[i]$).
אין צורך לכתוב את השיטה הזו!

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

אין צורך לדאוג ליעילות השיטה!

- מערך חד-ממדי vec באורך n נקרא **k-מאוזן** סביב אינדקס p כאשר $0 \leq p < n$ (על-פי ההגדרה בסעיף א) ואם התת-מערך $vec[0..p-1]$ ו- $vec[p..n-1]$ הם $-(k-1)$ -מאוזנים. (בסעיף ב אנו קוראים **מערך 1-מאוזן** למה שקראנו **מערך מאוזן** בסעיף א).
- מערך ריק הוא מאוזן. אם אין p שמאוזן, המערך נקרא 0-מאוזן.

לדוגמא, המערך $\{1, 2, 2, 1, 3, 2, 1, 3, 4, -1, 3, 3\}$ הוא **3-מאוזן** שכן:

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

- הוא מאוזן סביב $p=7$ $(1+2+2+1+3+2+1 = 3+4+-1+3+3)$
- כל אחד משני החלקים של המערך (אינדקסים 0 עד 6 ואינדקסים 7 עד 11) הוא 2-מאוזן.
- המערך $\{1, 2, 2, 1, 3, 2, 1\}$ הוא 2-מאוזן סביב $p=4$ $(1+2+2+1 = 3+2+1)$
- המערכים $\{1, 2, 2, 1\}$ ו- $\{3, 2, 1\}$ הם 1-מאוזנים. $(1+2 = 2+1, 3 = 2+1)$
- המערך $\{3, 4, -1, 3, 3\}$ הוא 2-מאוזן סביב $p=3$ (10 במערך המקורי) $(3+4+-1 = 3+3)$
- המערכים $\{3, 3\}$ ו- $\{3, 4, -1\}$ הם 1-מאוזנים. $(3 = 4+-1, 3 = 3)$

שימו לב, המערך להלן אינו 3-מאוזן:

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

שכן התת-מערך $\{4, 2\}$ באינדקסים 10 ו-11 אינו מאוזן. אבל הוא כן 2-מאוזן.

כתבו שיטה סטטית **רקורסיבית** `isBalanced`, המקבלת מערך `vec` ומספר אי-שלילי k , ובודקת אם `vec` הוא **k-מאוזן**.
חתימת השיטה היא:

```
public static boolean isBalanced (int [] vec, int k)
```

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

אין צורך לדאוג ליעילות השיטה! גם כאן כדאי להשתמש בהעמסת יתר (overloading). אפשר להשתמש בשיטה מסעיף א, גם אם לא כתבתם את המימוש שלה. אל תשכחו לתעד את מה שכתבתם!

שאלה 2 - 25 נקודות

a הוא מערך המכיל מספרים שלמים ממוין בסדר עולה **ממש** (כלומר המספרים שונים זה מזה).

נגדיר **מערך טווחים** של a כמערך שנבנה באופן הזה: עבור כל רצף של מספרים עוקבים ב-a, יהיה במערך הטווחים אובייקט שמכיל שני שדות של מספרים שלמים. שדה אחד הוא המספר הקטן ביותר ברצף (smallest) ושדה שני הוא המספר הגדול ביותר ברצף (largest). רצף יכול להיות באורך 1 או יותר. אם הרצף הוא באורך 1, הוא מיוצג במערך הטווחים על ידי אובייקט ששני השדות שלו שווים.

לדוגמא, עבור המערך a:

0	1	2	3	4	5	6	7	8	9	10
3	4	5	12	19	20	100	101	102	103	104

מערך הטווחים rangeA יהיה:

0	1	2	3
<3, 5>	<12, 12>	<19, 20>	<100, 104>

(כל אובייקט מסומן כזוג סדור עם סוגריים משולשים).

לשם כך הגדרנו מחלקה בשם Range בה כל אובייקט מייצג טווח של מספרים לפי המתואר לעיל.

להלן המחלקה Range:

```
public class Range
{
    private int _smallest, _largest;

    public Range(int sm, int la)
    {
        _smallest = sm;
        _largest = la;
    }

    public int getSmallest()
    {
        return _smallest;
    }

    public int getLargest()
    {
        return _largest;
    }
}
```

אפשר להניח שבכל אובייקט מהמחלקה _smallest תמיד קטן או שווה ל- _largest, ואין צורך לבדוק זאת.

עליכם לכתוב שיטה שמקבלת מערך טווחים ומספר x ומחזירה true אם אפשר למצוא זוג איברים שונים במערך המקורי שהסכום שלהם הוא x .

דוגמאות - בהינתן מערך הטווחים rangeA שלעיל:

- כאשר $x = 22$ השיטה תחזיר true (כי $3 + 19 = 22$)
- כאשר $x = 113$ השיטה תחזיר true (כי $12 + 101 = 113$)
- כאשר $x = 202$ השיטה תחזיר true (כי $100 + 102 = 202$)
- כאשר $x = 38$ השיטה תחזיר false (כי אין זוג מספרים במערך המקורי a שסכומם 38)
- כאשר $x = 53$ השיטה תחזיר false (כי אין זוג מספרים במערך המקורי a שסכומם 53)

חתימת השיטה היא:

```
public static boolean isSum(Range a[], int x)
```

אתם יכולים להניח שמערך הטווחים מייצג בצורה נכונה את המערך המקורי הממוין. אין צורך לבדוק זאת.

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.

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

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

א. נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
public class Node
{
    private char _data;
    private Node _leftSon, _rightSon;

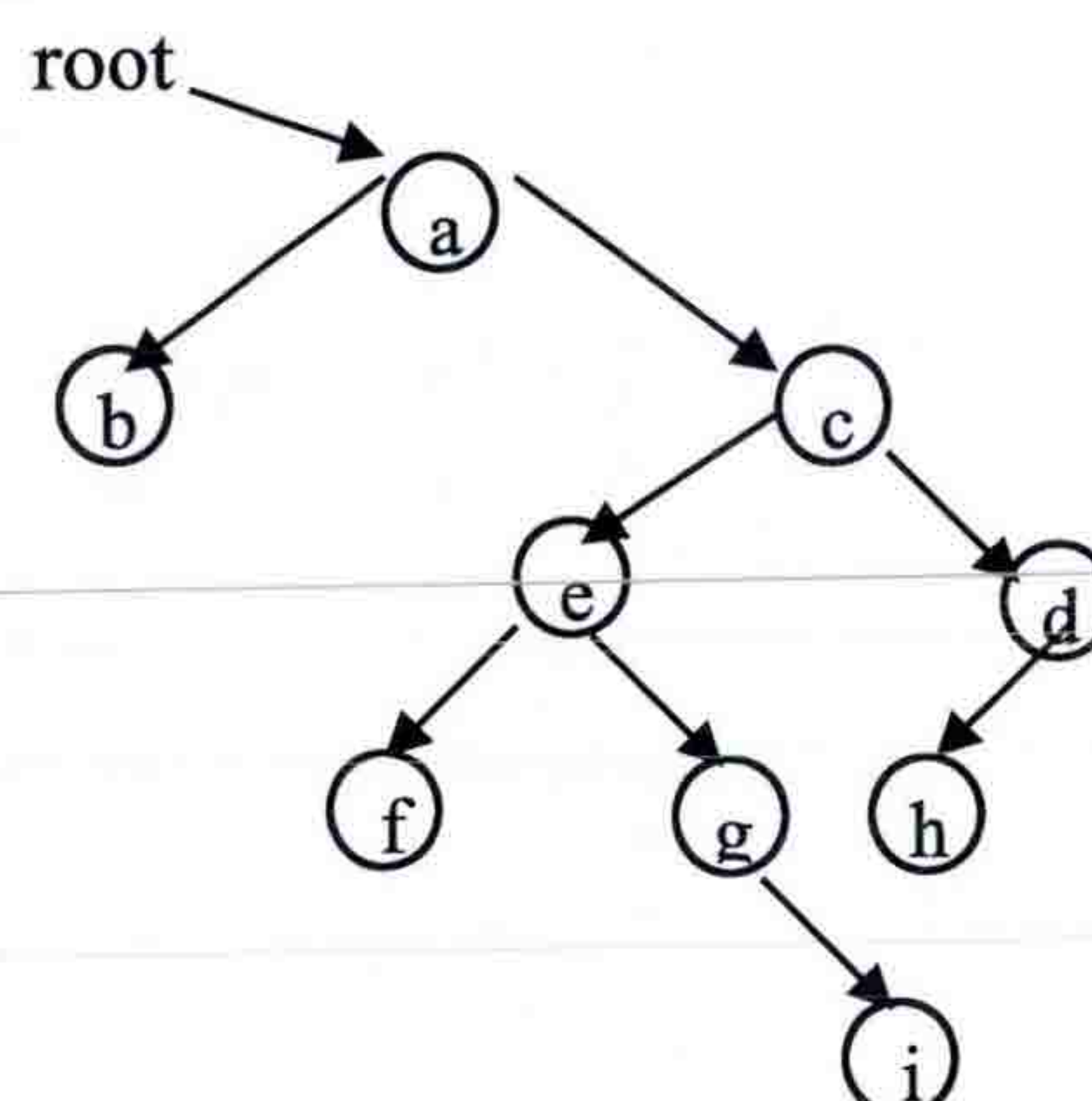
    public Node (char data)
    {
        _data = data;
        _leftSon = null;
        _rightSon = null;
    }

    public char  getData()           {return _data; }
    public Node getLeftSon()         {return _leftSon; }
    public Node getRightSon()        {return _rightSon; }
}
```

נתונה השיטה f הבאה שמקבלת כפרמטרים את t מטיפוס Node שמצביע לשורש של עץ בינרי ומחרוזת תווים st.

```
public static void f (Node t, String st)
{
    System.out.println (st);
    if (t.getLeftSon() != null)
        f ( t.getLeftSon() ,st + "0");
    if (t.getRightSon() != null)
        f (t.getRightSon(), st + "1");
}
```

נתון העץ הבינרי הבא:



ענו על שני הסעיפים הבאים:

(4 נק') (i) מה תדפיס השיטה f כאשר היא תקבל כמצביע את העץ $root$ הבא, ואת המחרוזת הריקה?

כלומר מה יודפס כתוצאה מביצוע הפקודה: $f(root, "")$;

התשובה היא:

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



(6 נק') (ii) מה מבצעת השיטה f באופן כללי? הסבירו בקצרה מה עושה השיטה 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. התייחסו למקרי הקצה השונים.

התשובה היא:



שאלה 4 - 12 נקודות

נתונות המחלקות הבאות:

```
public class AA{
    private char _ch;
    public AA(){
        _ch='+';
    }

    public AA(char ch){
        _ch=ch;
    }

    public char getCh(){
        return _ch;
    }
}

-----

public class BB extends AA{
    private String _st;
    public BB (){
        _st="BB";
    }

    public BB(String st, char ch){
        super(ch);
        _st=st;
    }

    public boolean equals (Object o)
    {
        System.out.print ("1 ");
        if ((o != null) && (o instanceof BB))
        {
            if (_st.equals(((BB)o)._st) &&
                (getCh() == ((BB)o).getCh()))
                return true;
        }
        return false;
    }

    public boolean equals (BB b)
    {
        System.out.print ("2 ");
        if (b != null)
        {
            if (_st.equals(b._st) &&
                (getCh() == b.getCh()))
                return true;
        }
        return false;
    }
}
```


ונתונה המחלקה Driver המשתמשת במחלקות AA ו-BB. כל המחלקות נמצאות באותה חבילה.

```
public class Driver
{
    public static void main (String [] args)
    {
        AA a1 = new AA();
        AA a2 = new BB();
        BB b1 = new BB();
        BB b2 = new BB();
        System.out.println(a2.equals(b1)); 1
        System.out.println(a2.equals(a1)); 2
        System.out.println(a1.equals(b1)); 3
        System.out.println(b1.equals(a1)); 4
        System.out.println(b1.equals(a2)); 5
        System.out.println(b1.equals(b2)); 6
    }
}
```

סעיף א – 6 נקודות

מה יהיה הפלט של השיטה main מהמחלקה Driver? שימו לב שבתחילת כל שיטת equals יש הדפסה של ספרה. אל תשכחו לכתוב גם אותה. התשובה היא:

סעיף ב – 6 נקודות

אם נוסיף למחלקה BB את השיטה הבאה:

```
public boolean equals (AA a)
{
    System.out.print ("3 ");
    if (a != null)
    {
        if (getCh() == a.getCh())
            return true;
    }
    return false;
}
```


מה יהיה הפלט של השיטה main מהמחלקה Driver? שימו לב שבתחילת כל שיטת equals יש הדפסה של ספרה. אל תשכחו לכתוב גם אותה. התשובה היא:

שאלה 5 - 16 נקודות

נתונה המחלקה IntNodeTwo הבאה, המייצגת איבר ברשימה מקושרת דו-סטריק בה יש מצביעים גם לאיבר הבא וגם לקודם, המכילה מספרים שלמים:

```
public class IntNodeTwo
{
    private int _num;
    private IntNodeTwo _next, _prev;

    public IntNodeTwo(int n) {
        _num = n;
        _next = null;
        _prev = null;
    }

    public IntNodeTwo(int num, IntNodeTwo n, IntNodeTwo p) {
        _num = num;
        _next = n;
        _prev = p;
    }

    public int getNum() { return _num; }
    public IntNodeTwo getNext() { return _next; }
    public IntNodeTwo getPrev() { return _prev; }
    public void setNum (int n) { _num = n; }
    public void setNext (IntNodeTwo node) { _next = node; }
    public void setPrev (IntNodeTwo node) { _prev = node; }
}
```

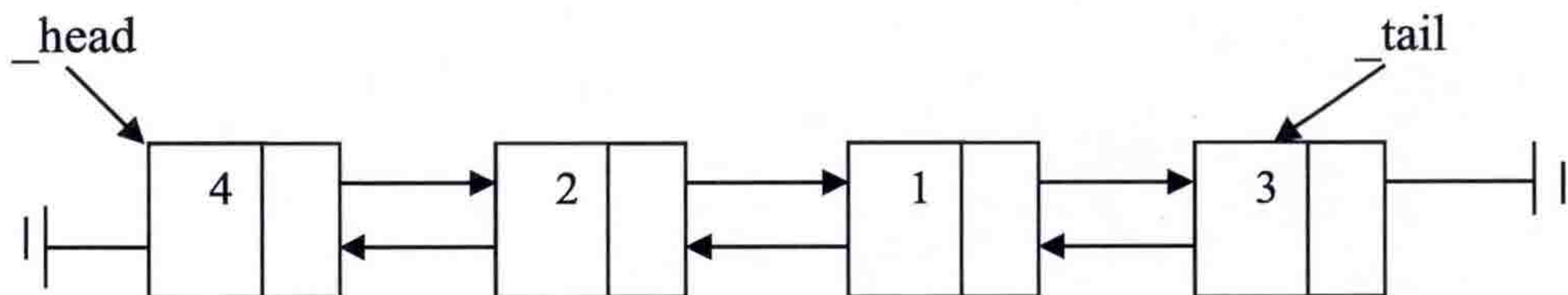

נתונה רשימה מקושרת דו-סטריק, הממומשת בעזרת המחלקה IntListTwo שלהלן:

```
public class IntListTwo
{
    private IntNodeTwo _head, _tail;
    public IntListTwo()
    {
        _head = null;
        _tail = null;
    }

    // כאן יש עוד בנאים ושיטות...
    public void secret()
    {
        IntNodeTwo p1 = null;
        IntNodeTwo p2 = _head;
        IntNodeTwo p3 = null;
        int temp;
        do{
            if (p2.getNum()%3 == 1)
                p2 = p2.getNext();
            else
                if (p2.getNum() %3 == 0)
                {
                    if (p1 == null)
                        p1 = _head;
                    else
                        p1 = p1.getNext();
                    temp = p1.getNum();
                    p1.setNum(p2.getNum());
                    p2.setNum(temp);
                    p2 = p2.getNext();
                }
            else
            {
                if (p3 == null)
                    p3 = _tail;
                else
                    p3 = p3.getPrev();
                temp = p3.getNum();
                p3.setNum(p2.getNum());
                p2.setNum(temp);
            }
        }
        while (!(p2==p3) || (p3!=null && p3.getNext()==p2));
    }
} //end of class IntListTwo
```


הניחו שיש במחלקה גם שיטה שמכניסה ערכים לרשימה. אינכם צריכים לדאוג לכך.

בשאלות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים. כך לדוגמא, נסמן $\{ 4, 2, 1, 3 \}$ את הרשימה שלהלן (המספר הראשון משמאל הוא המספר שבראש הרשימה:



סעיף א: (6 נקודות)

איך תיראה הרשימה $\{ 2, 4, 6, 1, -3, 0, 2, 0, -4, 5, 3 \}$ לאחר שנפעיל עליה את השיטה secret? כתבו את האיברים ברשימה בתוך סוגריים כמו שאנחנו רשמנו (משמאל לימין).

X

סעיף ב: (10 נקודות)

מה מבצעת השיטה secret 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 איד היא מבצעת זאת.

שאלה 6- 12 נקודות

עבור כל אחת מהפעולות שבטבלה, ציינו מהו סדר גודל זמן הריצה הדרוש כדי לבצע את הפעולה במערך ממוין בסדר עולה (מהקטן לגדול) ובמערך שאינו ממוין. כלומר, עליכם להתייחס לאלגוריתם יעיל ככל האפשר ולכתוב מה סדר גודל זמן הריצה שלו. כל זאת ללא שימוש במערכי עזר, כלומר בסיבוכיות מקום קבועה $O(1)$, ובלי לשנות את אופי המערכים (ממוין/לא ממוין), ומבלי להשאיר חורים במערך. גודל המערך הוא n .

בשני המקרים ניתן להניח ששום פעולה לא תגרום לחריגה מגבולות המערך.

הפעולה	זמן במערך ממוין	זמן במערך לא ממוין
הכנסת איבר למערך	X	✓
חישוב סכום האיבר הרביעי בגודלו עם האיבר השביעי בגודלו	✓	X
בדיקה אם איבר x נמצא במערך	X	✓
הוצאת המינימום מהמערך	X	X
בדיקה אם יש זוג איברים במערך שסכומם שווה למספר כלשהו x .	✓	✓
בדיקה מה ההפרש הגדול ביותר בין שני איברים כלשהם במערך	✓	X

שימו לב, אינכם צריכים לכתוב את האלגוריתם אליו אתם מתכוונים אלא רק את הסיבוכיות של האלגוריתם היעיל ביותר.

בהצלחה

