

מטלת מחשב (ממ"ח) 03

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 19 - 20 נושא המטלה: תורים, מחסניות, עצים בינריים

מספר השאלות: 20 משקל המטלה: 3 נקודות

סמסטר: 2015 מועד אחרון להגשה: 10.2.2015

(ת)

את התשובות לממ"ח יש לשלוח באמצעות מערכת שאילת"א

בכתובת <http://www.openu.ac.il/sheilta/>

הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח למנחה!

בשאלות מטלה זו, יש להעזר בממשקים למחלקות Queue, Stack, Node
הנתונים בהרצאות הקורס.

שאלה 1. איזה מהמשפטים הבאים נכון?

- אם נממש את המחלקה Stack בעזרת רשימה מקושרת, אזי הדרך היעילה ביותר לממש את הפעולה `empty()`, שמחזירה אמת אם המחסנית ריקה, היא ב $O(n)$.
- אין הבדל ממשי בין המחלקות Stack ו Queue וניתן לבחור להשתמש בכל אחת מהן באופן אקראי.
- FIFO הוא מושג שמתאר התנהגות של תור – הראשון שנכנס הוא הראשון שיוצא.
- FIFO הוא מושג שמתאר התנהגות של מחסנית – הראשון שנכנס הוא הראשון שיוצא.

שאלה 2. הניחו שיש תור q (מהמחלקה Queue) שמכיל את המספרים 1,2,3,4,5,6 בסדר זה

(המספר 1 בראש התור). הניחו שנתונה s מחסנית ריקה ולא ניתן לייצר עוד מחסניות ותורים, וניתן לבצע רק שלושה סוגים של פעולות:

- i. להוציא איבר מ-q (`dequeue`) ולהדפיס אותו
- ii. להוציא איבר מ-q (`dequeue`) ולשים אותו ב-s (`push`)
- iii. להוציא איבר מ-s (`pop`) ולהדפיס אותו.

איזה מהפלטים הבאים (משמאל לימין) אינו אפשרי בשימוש אך ורק בפעולות אלה?

א. 123456

ב. 654321

ג. 234561

ד. 135246

ה. 125643

שאלה 3. מה מבצעת השיטה הבאה:

```
public void something()
{
    Stack s = new Stack();
    while (!empty())
    {
        int num = pop();
        System.out.print (num + "\t");
        s.push(num);
    }
    System.out.println();
    while (!s.empty())
    {
        int num = s.pop();
        push(num);
    }
}
```

א. מדפיסה את הערכים שבמחסנית מראש המחסנית אל זנבה

ב. מדפיסה את הערכים שבמחסנית מזנב המחסנית אל ראשה

ג. מדפיסה את הערכים שבמחסנית מראש המחסנית אל זנבה ובחזרה מהזנב אל הראש

ד. מדפיסה את הערכים שבמחסנית מזנב המחסנית אל ראשה ובחזרה מהראש אל הזנב

שאלה 4.

נניח ש $s1$ ו- $s2$ הן שתי מחסניות מהמחלקה Stack כאשר $s1$ מלאה ב- n איברים (אפשר להניח ש n מספר גדול) ו- $s2$ היא מחסנית ריקה. x הוא משתנה מסוג `int`. ידוע גם שקיימת שיטה בשם `length` שמקבלת כפרמטר אובייקט מהמחלקה Stack ומחזירה את אורך המחסנית.

בהינתן קטע הקוד הבא:

```
int len = length(s1) - 3;
for (int i = 0; i < len; i++){
    x = s1.pop();
    s2.push(x);
}

len = length(s1) - 3;
for (int i = 1; i <= len; i++){
    x = s2.pop();
    s1.push(x);
}
```

לאחר הרצת קטע הקוד שלעיל, מה מהמשפטים הבאים נכון?

- א. $s1$ לא תשתנה, הערך של x יהיה שווה לערך ה-4 מהתחתית של המחסנית s
- ב. $s1$ לא תשתנה והערך של x יהיה זהה לערך שבראש המחסנית $s1$
- ג. $s1$ תכלול שלושה איברים והערך של x יהיה זהה לערך שבראש המחסנית $s1$
- ד. $s1$ לא תשתנה והערך של x יהיה זהה לערך שבראש המחסנית $s2$
- ה. $s1$ תכלול שלושה איברים והערך של x יהיה זהה לערך בתחתית המחסנית $s1$
- ו. $s1$ תכלול שלושה איברים והערך של x יהיה זהה לערך שבראש המחסנית $s2$

```

public void secret(){
    _head = secret(this, new Stack(),null)._head;
}

private Stack secret(Stack source, Stack dest, Stack s){
    if (s==null)
        s = new Stack();

    if (!source.empty()){
        int temp = source.pop();
        while (!source.empty()){
            if (source.top()<temp){
                dest.push(temp);
                temp = source.pop();
            }
            else{
                dest.push(source.pop());
            }
        }
        s.push(temp);
        return secret(dest,source,s);
    }
    else {
        return s;
    }
}

```

בהנחה שהמחסנית עליה מתבצעת השיטה secret הכילה את האיברים הבאים:
(משמאל לימין, בשמאל ראש המחסנית)

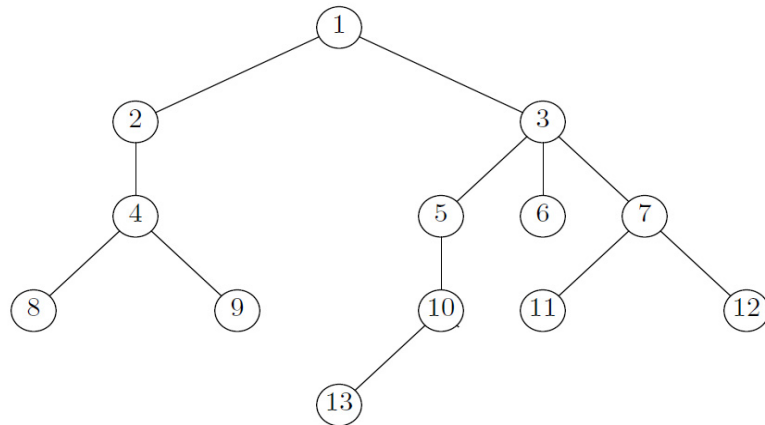
5 2 4 3 7 6

_head הוא ראש המחסנית.

איך תיראה המחסנית לאחר ביצוע השיטה secret ? (משמאל לימין, בשמאל ראש המחסנית)

- | | | | | | | |
|----|---|---|---|---|---|---|
| א. | 6 | 7 | 3 | 4 | 2 | 5 |
| ב. | 2 | 3 | 4 | 5 | 6 | 7 |
| ג. | 7 | 6 | 5 | 4 | 3 | 2 |
| ד. | 5 | 6 | 2 | 7 | 4 | 3 |

שאלה 6. נתון עץ ששורשו הוא הצומת 1 :



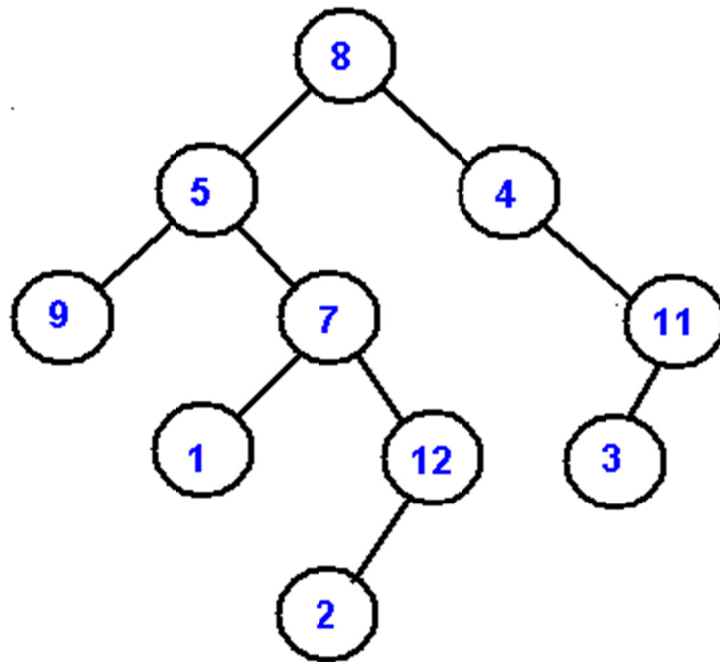
נתונות הטענות הבאות :

1. השורש של העץ הוא : 1
2. העלים של העץ הם : 13, 12, 11, 9, 8, 6
3. הצמתים הפנימיים של העץ הם : 10, 7, 5, 4, 3, 2, **1**
4. האבא של הצומת 6 הוא 3
5. 6 הוא בן של 3
6. 3 הוא אב קדמון של 12
7. 3 הוא אב קדמון של 13
8. 3 הוא אבא של 5, 6, 7

סמנו את התשובה הנכונה :

- א. כל הטענות נכונות
- ב. רק הטענות 1, 2, 3, 4, 5 ו-6 נכונות
- ג. רק הטענה 6 אינה נכונה
- ד. רק הטענות 1, 2, 3, 4, 5 ו-7 נכונות
- ה. אף תשובה אינה נכונה

שאלה 7. נתון העץ הבינרי הבא



נתונות הטענות הבאות :

1. סדר הביקור בצמתים של העץ לאחר סיור InOrder הוא 8, 5, 9, 7, 1, 12, 2, 4, 11, 3
2. סדר הביקור בצמתים של העץ לאחר סיור InOrder הוא 9, 5, 1, 7, 2, 12, 8, 4, 3, 11
3. סדר הביקור בצמתים של העץ לאחר סיור PostOrder הוא 9, 1, 2, 12, 7, 5, 3, 11, 4, 8
4. סדר הביקור בצמתים של העץ לאחר סיור PostOrder הוא 3, 11, 4, 2, 12, 1, 7, 9, 5, 8
5. סדר הביקור בצמתים של העץ לאחר סיור PreOrder הוא 8, 5, 9, 7, 1, 12, 2, 4, 11, 3
6. סדר הביקור בצמתים של העץ לאחר סיור PreOrder הוא 8, 5, 4, 9, 7, 11, 1, 12, 3, 2

סמנו את התשובה הנכונה :

- א. הטענות 2, 3, 6 נכונות
- ב. הטענות 2, 3, 5 נכונות
- ג. רק הטענות 2 ו-3 נכונות
- ד. אף תשובה אינה נכונה

שאלה 8. לאחר הפעלת סיור PreOrder ו InOrder על עץ בינארי כלשהו התקבלו התוצאות הבאות:

InOrder : 2 6 4 7 1 3 8 5 9 10

PreOrder : 1 2 4 6 7 3 5 8 9 10

סמנו את התשובה הנכונה :

- א. בהסתמך על תוצאת הסיור **InOrder** ו **PreOrder** ניתן לשחזר את העץ המקורי
- ב. לא ניתן לשחזר את העץ המקורי על סמך תוצאת הסיורים שהתקבלה
- ג. ניתן לשחזר את העץ המקורי אך ורק לו והיה בידינו גם את תוצאת הסריקה לפי סיור PostOrder
- ד. בהסתמך על התוצאות ניתן לבנות עץ בינארי כלשהו אבל לא ניתן להבטיח שזהו העץ המקורי

שאלה 9. לאחר הפעלת סיור PreOrder על עץ **חיפוש** בינארי כלשהו התקבלה התוצאה הבאה:

PreOrder : 5 1 4 10 7 6 8 13

אם נפעיל סיור PostOrder על העץ הזה נקבל את התוצאה הבאה.

סמנו את התשובה הנכונה :

- א. 1 4 5 6 7 8 10 13
- ב. 13 8 6 7 10 4 1 5
- ג. 4 1 6 8 7 13 10 5
- ד. אי אפשר לדעת מה היה העץ המקורי על סמך תוצאת סיור ה-PreOrder ולכן לא ניתן לדעת מה תוצאת סיור ה- PostOrder על העץ.

קטע הקוד הבא מתייחס לשאלות 10 ו-11.

נתון עץ בינארי כלשהו T בעל n צמתים שמכילים מספרים שלמים חיוביים בלבד
כמו כן נתון קטע הקוד הבא :

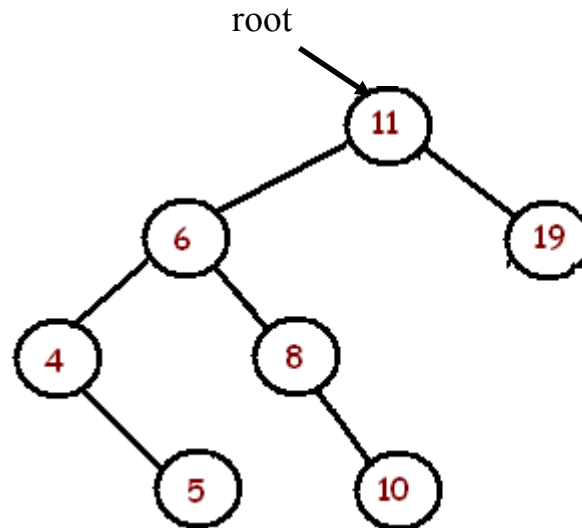
```
public int what(Node root)
{
    if(root == null)
        return 0;
    int x = what(root.getLeft());
    int y = what(root.getRight());
    return root.getValue() + f(x,y);
}

private int f(int x,int y)
{
    return y>x? y:x;
}
```

השיטה what מקבלת כפרמטר מצביע לשורש העץ T.

שאלה 10.

מה תחזיר השיטה what לעיל בהינתן לה העץ הבינרי הבא ששורשו root?



סמנו את התשובה הנכונה

- א. 30
- ב. 19
- ג. 21
- ד. 35
- ה. 63

שאלה 11.

סמנו את התשובה הנכונה :

- א. השיטה what מחזירה את המסלול הארוך ביותר בעץ
- ב. השיטה what מחזירה את הסכום של האיברים בעץ לאורך המסלול הארוך ביותר
- ג. השיטה what מאתרת את המסלול מהשורש לעלה אשר סכום הצמתים לאורכו הוא מקסימלי ומחזירה את הסכום
- ד. התשובות ב ו- ג נכונות

הקטע הבא (ובו קטעי מחלקות) מתייחס לשאלות 12-15

עץ טרנארי (Ternary tree) הוא עץ שיש בו לכל היותר שלושה בנים לצומת: בן שמאלי, בן אמצעי ובן ימני.

נניח שהמחלקה TerNode שלהלן מממשת צומת בעץ טרנארי.

```
public class TerNode
{
    private int _number;
    private TerNode _leftSon, _middleSon, _rightSon;

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

    public int getNumber()          {return _number;}
    public TerNode getLeftSon()     {return _leftSon;}
    public TerNode getMiddleSon()   {return _middleSon;}
    public TerNode getRightSon()    {return _rightSon;}

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

המחלקה TernaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ טרנארי.

בין השיטות נתונות השיטות what ו-something הבאות:

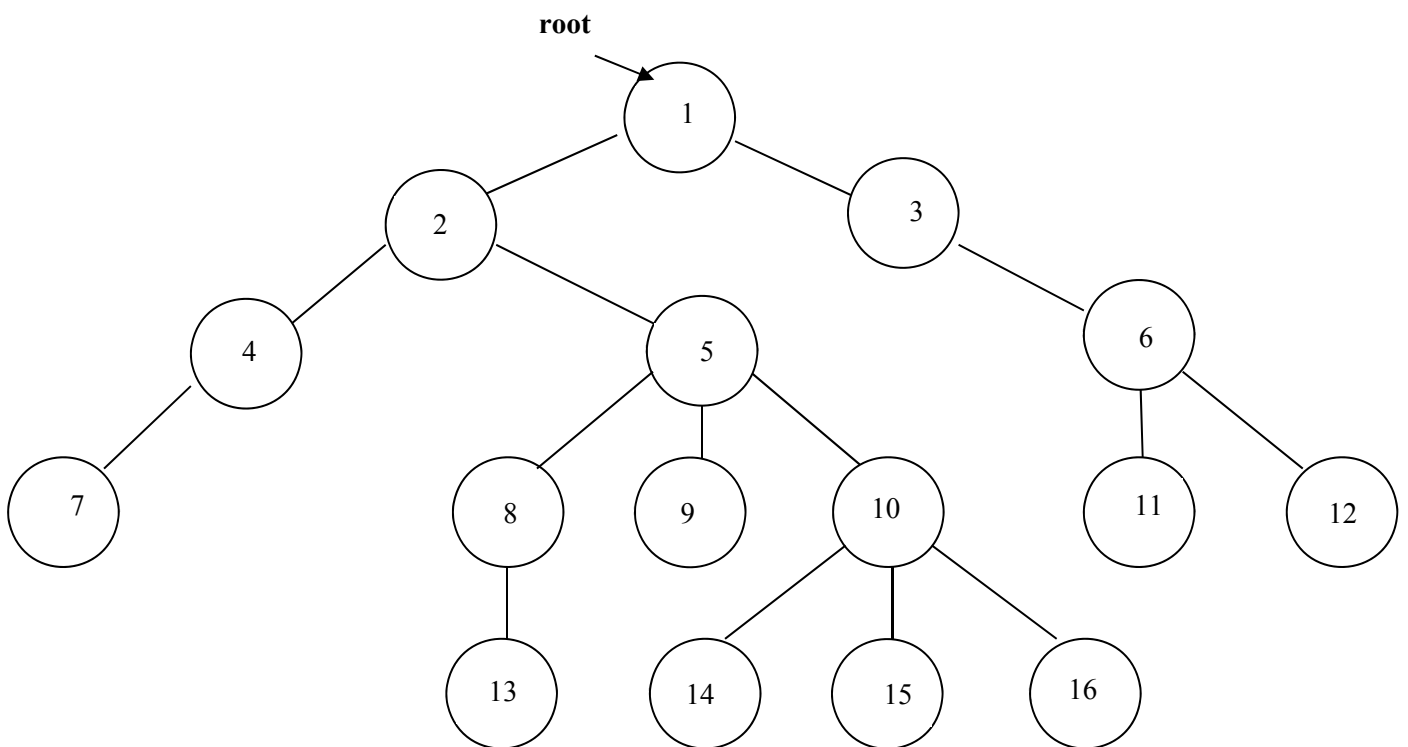
```
public static int what (TerNode t)
{
    if (t == null)
        return 0;
    int temp = 0;
    if (t.getLeftSon() != null) temp++;
    if (t.getMiddleSon() != null) temp++;
    if (t.getRightSon() != null) temp++;
    return temp;
}
```

```

public static boolean something (TerNode t)
{
    if (t == null)
        return true;
    if (what(t) == 3)
        return false;
    return something (t.getLeftSon())
        && something (t.getMiddleSon())
        && something (t.getRightSon());
}

```

נתון העץ הטרינארי הבא, ששורשו הוא root



שאלה 12.

- מה מבצעת השיטה what באופן כללי, בהינתן לה צומת t של עץ טרינארי?
- השיטה מחזירה את מספר הבנים המקסימלי של צומת בעץ ששורשו t.
 - השיטה מקבלת צומת t ומחזירה את מספר בניו (הישירים).
 - השיטה מקבלת צומת t ומחזירה את מספר צאצאיו (כולל בניו הישירים).
 - השיטה מקבלת צומת t ומחזירה את מספר צאצאיו (לא כולל בניו הישירים).

שאלה 13.

בהינתן העץ הטרנארי לעיל ששורשו root, איזה ערך תחזיר השיטה something בעקבות הקריאה

`?BinaryTree.something(root)`

- א. 3
- ב. 1
- ג. true
- ד. false

שאלה 14.

אם נרצה שהשיטה something תחזיר תשובה אחרת מזו שהוחזרה בשאלה 13, אילו שינויים **מינימליים** עלינו לבצע **בעץ** לעיל? (כלומר, שינויים שיגעו במינימום צמתים)

- א. יש למחוק את הצמתים 9 ו-15 מהעץ
- ב. יש למחוק את הצמתים 8 ו-15 מהעץ
- ג. יש למחוק את הצומת 10 (**ובניו**) מהעץ
- ד. יש להוסיף צומת כבן של הצומת 6.

שאלה 15.

מה מבצעת השיטה something באופן כללי, בהינתן לה צומת t של עץ טרנארי?

- א. השיטה בודקת מה מספר הבנים של כל צומת בעץ ששורשו t, ומחזירה true אם הוא 3 ו- false אם הוא קטן מ-3.
- ב. השיטה בודקת מה מספר הבנים של כל צומת בעץ ששורשו t, ומחזירה false אם הוא 3 ו- true אם הוא גדול מ-3.
- ג. השיטה בודקת אם הצומת t הוא עלה ומחזירה true אם כן, ו- false אם לא.
- ד. השיטה בודקת אם הצומת t הוא שורש של עץ בינרי ומחזירה true אם כן ו- false אם לא.

שאלה 16. נתונה השיטה הבאה:

```
public static boolean secret(Node t)
{
    if (t.getLeft()==null && t.getRight()==null)
        return true;
    if (t.getLeft()==null || t.getRight()==null)
        return false;
    if (t.getLeft().getInfo()>0)
        return false;
    if (t.getRight().getInfo()<0)
        return false;
    return secret(t.getLeft()) && secret (t.getRight());
}
```

נתונות שתי הטענות הבאות:

1. **קיים** עץ t המכיל רק ערכים חיוביים (גדולים מ-0), כך שהשיטה $\text{secret}(t)$ תחזיר true.
2. **לא קיים** עץ t שלא מכיל ערכים חיוביים, כך שפעולה $\text{secret}(t)$ תחזיר true.

הנחות לשתי הטענות:

1. ב- t יש לפחות שני צמתים.
2. צומת לא יכול להכיל את הערך 0.

סמנו את התשובה הנכונה:

- א. שתי הטענות לא נכונות.
- ב. טענה 1 נכונה וטענה 2 לא נכונה.
- ג. טענה 1 לא נכונה וטענה 2 נכונה.
- ד. שתי הטענות נכונות.

שאלה 17. בהינתן עץ חיפוש בינרי, איזה מהמשפטים הבאים נכון?

- א. אם רוצים להוסיף איבר מסוים x לעץ יתכן ויש יותר ממקום אחד שמתאים לו בעץ כך שעדיין העץ ימשיך להיות עץ חיפוש בינרי
- ב. אם נתקדם בעץ כל פעם שמאלה נגיע לאיבר המקסימלי בעץ
- ג. שורש העץ הוא בהכרח האיבר המינימלי בעץ
- ד. בהנתן רשימת ערכים x_1, x_2, \dots, x_n , סדר הכנסת האיברים לעץ משפיע על מבנה העץ. כלומר, אם נכניס את האיברים בסדר שונה, מבנה העץ יהיה שונה.

שאלה 18. איזה משפט מהמשפטים הבאים אינו נכון?

- א. בבסיס בינרי 10 ועוד 10 שווה 100.
- ב. 8 ביט נקראים בייט (byte).
- ג. 8 ביט זה גודל מספיק כדי לייצג מספר בין 0 ל – 127.
- ד. 8 ביט זה גודל מספיק כדי לייצג מספר בין 0 ל – 255.
- ה. 8 ביט זה גודל מספיק כדי לייצג מספר בין 0 ל – 511.

שאלה 19. מה ערך המספר הבינרי 0011101101 בבסיס עשרוני (10) ?

- א. 6
- ב. 235
- ג. 109
- ד. 786
- ה. אף אחת מהתשובות א-ד אינה נכונה

שאלה 20. אם נקח מספר בבסיס בינרי (2), ונוסיף ארבעה אפסים בסוף המספר, איזה משפט

מהמשפטים הבאים נכון?

- א. המספר העשרוני יוכפל ב-10000.
- ב. המספר העשרוני יוכפל ב-4.
- ג. המספר העשרוני יוכפל ב-8.
- ד. המספר העשרוני יוכפל ב-16.
- ה. אין קשר בין המספר המקורי למספר החדש.

ב ה צ ל ח ה