67101
תשס"ז/5

האוניברסיטה העברית בירושלים
המכון למדעי המחשב

<u>בחינה במבוא למדעי המחשב</u>
קורס מס' 67101

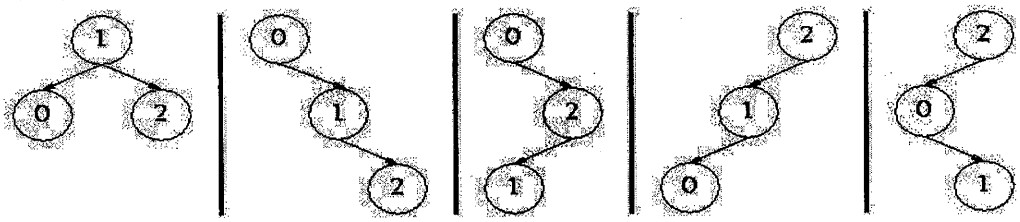
מועד ב' תש"ס
המורה: פרופ' ג'ף רוזנשיין

תאריך: 18.7.00
זמן: 2 ½ שעות

ענה על שלוש שאלות מתוך ארבע.

1. עליכם לממש פונקציה רקורסיבית המקבלת מספר שלם אי שלילי - n - ומחזירה את מספרם של עצי החיפוש הבינריים האפשריים עבור n קודקודים שונים.
לדוגמא, עבור n=3 הפונקציה תחזיר 5 ...

all possible binary search trees on 3 different vertices:



מגדירים שעבור n=0 הפונקציה תחזיר 1 (העץ הריק).

2/...

67,101
תשס"ז

2. א. (28 נק') לרשותכם המחלקות הבאות:

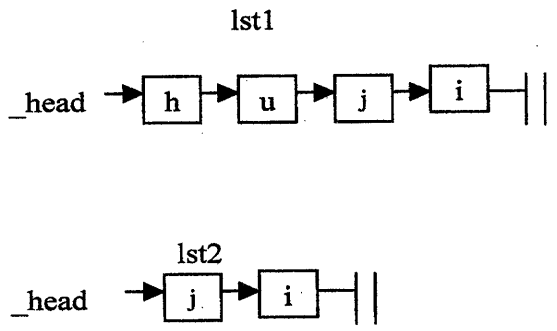
```
public class CharNode {
  private char _data;
  private CharNode _next;

public CharNode(char  data) {
    _data = data;
    _next = null;
  }
public void addNext(CharNode node) {_next = node;}
public CharNode getNext() {return _next;}
public char getData() {return _data;}
public void removeNext() {
    CharNode tmp = getNext();
    if(tmp != null) {
      addNext(tmp.getNext());
      tmp._next = null;
    }
  }
}
```

```
public class CharList {
  private CharNode _head;

  public CharList() {
    _head = null;
  }
  public CharNode getHead() {return(_head);}
  public void add(char data) {
    CharNode tmp;
    CharNode newNode = new CharNode(data);

    if(_head == null)
      _head = newNode;
    else {
      tmp = _head;
      while(tmp.getNext()!=null)
            tmp = tmp.getNext();
      tmp.addNext(newNode);
    }
  }
  public int firstAppearance(CharList other) {}
}
```

כתבו שיטה שכותרתה public int firstAppearance(CharList other) אשר תחזיר את האינדקס של המופע הראשון של other בתוך הרשימה הנוכחית ואם אין מופע כזה תחזיר 1– .

<u>הערה</u>: שימו לב כי המחרוזת הריקה מופיעה באינדקס 0 של כל מחרוזת אחרת.

דוגמא:



הקריאה lst1.firstAppearance(lst2) תחזיר 2.

ב. (5 נק') מה הסיבוכיות של השיטה שמימשתם?

3/14

67,101
תשס"ב

3. נתונה התוכנית הבאה:

```
class A {
    public int _a = 7;
    void foo() {
        B b = new B(this);
        b.foo();
    }
    void print() {
        B b = new B(this);
        System.out.println(_a+","+b._b);
    }
}

class B {
    private A _a;
    public int _b = 7;
    public B(A a) {
        _a = a;
    }
    public void foo() {
        _a._a++;
        _b = _a._a;
    }
}

class Main {
    static public void main(String[] args){
        A a = new A();
        a.foo();
        a.print();
    }
}
```

א. מה יהיה הפלט שלה ומדוע ?

ב. מה יהיה פלט התוכנית אם נצהיר על המישתנה _b של המחלקה B כמישתנה סטטי:

```
static public int _b;
```

הסבירו מדוע.

ג. אם היינו הופכים לפרטיים את המשתנה _a של מחלקה A ומישתנה _b של מחלקה B:

```
private int _a;
private int _b;
```

איזה טעויות קומפילציה היינו מקבלים ולאיזה שורות בקוד הן היו מתייחסות ?

4/111

67.101
תש"מ/ב

4. כתבו פונקציה:

```
public void pairsNum(int[] myArg) { ... }
```

אשר:

א. עוברת על רצף ה- integers הנתון ע"י myArg וסופרת את מספר המופעים של כל זוג integers עוקבים (עוקבים מבחינת הסדר שלהם במערך).

ב. אחרי המעבר הנ"ל, הפונקציה תדפיס בעבור חמשת הזוגות השכיחים ביותר את:

- שני ה- integers שמרכיבים את הזוג.
- מספר הפעמים שזוג זה הופיע.
- אחוז המופעים של אותו זוג מתוך מספר כל הזוגות (העוקבים) במערך.

ג. סדר ההדפסה מוגדר כך: אם מספר המופעים של שני זוגות שווה, אין זה משנה איזה זוג יודפס קודם, אולם אם זוג <i1,i2> מופיע יותר פעמים מזוג <j1,j2> אזי הזוג <i1,i2> יודפס קודם.

הנחה:
כל המספרים במערך myArg הינם בתחום: 0,1,2,3,4,5,6,7,8,9 (כלומר ניתן לייצגם ע"י ספרה עשרונית אחת).

דוגמא:
נניח myArg נתון ע"י:
7,2,9,8,3,0,1,8,7,4,6,2,9,3,8,3,0,1,2,9,1,8,3,9,2,8,4,7,1,9,9
אזי הזוגות העוקבים הם:
<7,2> <2,9> <9,8> **<8,3>** <3,0> <0,1> <1,8> <8,7> <7,4> <4,6> <6,2> <2,9> <9,3>
<3,8> **<8,3>** <3,0> <0,1> <1,2> <2,9> <9,1> <1,8> **<8,3>** <3,9> <9,2> <2,8> <8,4>
<4,7> <7,1> <1,9> <9,9>

הזוג <2,9> מופיע 3 פעמים (10% מכלל הזוגות)
הזוג <8,3> מופיע 3 פעמים (10% מכלל הזוגות)
הזוג <3,0> מופיע 2 פעמים (7% מכלל הזוגות)
וכן הלאה ...

הנחיות:
עליכם להשתמש במבני נתונים שיאפשרו ספירה (יעילה) של הזוגות וכן הדפסה ממוינת (יעילה) של חמשת הזוגות השכיחים (כפי שתואר למעלה).
מותר (ואף רצוי) להשתמש בפונקציות עזר.

בהצלחה!