מבוא למדעי המחשב
202-1-101-1

מבוא לתכנות למערכות מידע
202-1-104-1

סמסטר א', תשס"ז, 2006/7
בחינת מועד ב' – 16.2.2007

ד"ר ג'יהאד אל-סאנע
מר איתי בר-יוסף
גב' דקלה דותן
פרופ' מיכאל קודיש
ד"ר צחי רוזן
מר גיא שני

משך הבחינה: שעתיים וחצי.
חומר עזר: אסור.

בבחינה זו 4 שאלות. ענו על כולן. הניקוד מסתכם ב-100 נקודות.

בשאלות התכנות מספר השורות העומדות לרשותכם בדף התשובות אינו רומז על אורך הקוד הנדרש. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא. מותר להסתמך על סעיפים קודמים גם אם לא עניתם עליהם.

שימו לב: בשאלות התכנות החשיבות העליונה היא לנכונות הקוד, מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגננת תזכה בציון גבוה יותר.

רשמו את תשובותיכם במקומות המיועדים לכך בטופס הבחינה בלבד. המחברת שקיבלתם היא מחברת טיוטה ולא תימסר כלל לבדיקה.

בהצלחה!

שאלה 1 (20 נקודות)

תור עדיפויות הוא תור "לא צודק" שבו סדר היציאה של העצמים אינו תלוי בסדר הכניסה, אלא בעדיפות שלהם. תור עדיפויות תומך בפעולת הכנסה של אובייקט ובפעולת הוצאה של האובייקט עם הערך המקסימלי. האובייקטים אשר מוכנסים לתור העדיפויות הם מסוג Comparable, ופעולת ההשוואה ביניהם תקבע את סדר העדיפויות.

```
public interface Comparable{
        public int compareTo(Object oOther);
}
```

בשאלה זו נניח כי אין שני אובייקטים עם אותו סדר עדיפות.

נתון הממשק הבא:

```
public interface PriorityQueue{
        public boolean isEmpty();
        public void insert(Comparable cData);
        public Object extractMax();
}
```

הניחו כי קיימת מחלקה AStack המממשת את הממשק Stack:

```
public interface Stack{
        public boolean isEmpty();
        public void push(Object oData);
        public Object pop();
}
```

עליכם לממש את המחלקה PriorityQueueAsStack אשר מממשת את הממשק PriorityQueue באמצעות מחסנית.

רמז: יש לשמור על האובייקטים במחסנית שיהיו ממוינים בכל שלב.

```
public class PriorityQueueAsStack implements PriorityQueue {
        Stack m_sPQ;

        public PriorityQueueAsStack(){
                m_sPQ =new AStack();
        }
```

<u>סעיף א' (5 נקודות)</u>

ממשו את השיטה isEmpty אשר בודקת אם התור ריק.

```
        public boolean isEmpty(){

                _____________________________________
                        השלימו בדף התשובות
                _____________________________________


        }
        public Object extractMax(){
                return m_sPQ.pop();
        }
```

<u>סעיף ב' (15 נקודות)</u>

ממשו את השיטה insert אשר מוסיפה איבר חדש לתור.

רמז: ניתן להשתמש במחסנית עזר.

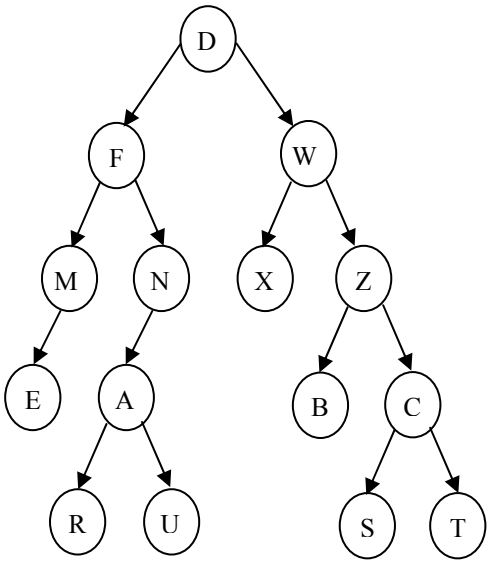
```
        public void insert(Comparable cData){

                _____________________________________
                        השלימו בדף התשובות
                _____________________________________


        }
}//class
```

שאלה 2 (30 נקודות)

נתונה המחלקה BinaryTree כפי שנלמדה בכיתה (הקוד מצורף בנספח הבחינה). עומק (depth) של קודקוד מוגדר כמרחק שלו מהשורש. העומק של קודקוד D הוא 0 והעומק של קודקוד A הוא 3 (ראה איור 1). נגדיר את רמה (Level) L כרשימת הנתונים (data) של הקודקודים באותו עומק (ראה דוגמה ל- Level 3 באיור 2).



איור -1

סעיף א' (10 נקודות)

השלימו את השיטה הפרטית computeLevelList(…) במחלקה BinaryTree המחשבת ומכניסה לרשימה מקושרת (LinkedList) את כל הנתונים (data) של הקודקודים באותו עומק (ראה איור 2).
שים לב שהשיטה הציבורית computeLevelList(…) קוראת לפרטית.

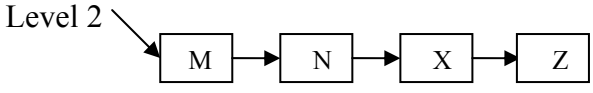
```
public LinkedList computeLevelList(int iDepth){
        LinkedList llLevel = new LinkedList();
        if ( m_bnRoot != null )
                computeLevelList(llLevel, iDepth, m_bnRoot);
        return llLevel ;
}

private void computeLevelList(LinkedList llLevel, int iDepth, BinaryNode bnNode){

        _____________________________________
                השלימו בדף התשובות
        _____________________________________

}
```

שים לב שאין חשיבות לסדר הנתונים ברשימה.
הערה: ניתן להניח כי הפרמטרים חוקיים.



איור -2

<u>סעיף ב' (10 נקודות)</u>

השלימו את המחלקה LevelIterator שממשת את הממשק Iterator באופן הבא: בכל קריאה ל-next() תוחזר רשימה מקושרת (LinkedList) הכוללת את נתונים (data) של הקודקודים באותה רמה. בקריאה הראשונה תוחזר רשימה המכילה את השורש בלבד, ובקריאה השלישית next() תחזיר את הרשימה Level 2 באיור 2.

שים לב שאין חשיבות לסדר הנתונים ברשימה.
הערה: ניתן להשתמש בשיטה height() במחלקה BinaryTree אשר מחשבת את גובהו של העץ(עומק מקסימאלי).

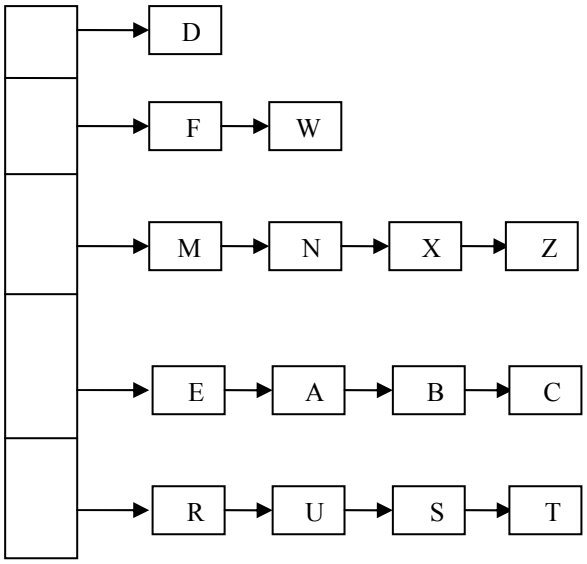
```
public class LevelIterator implements Iterator{
        _____________________________________
                השלימו בדף התשובות
        _____________________________________

        public  LevelIterator( _______________ ){
                _____________________________________
                השלימו בדף התשובות
                _____________________________________
        }
        public boolean hasNext(){
                _____________________________________
                השלימו בדף התשובות
                _____________________________________
        }
        public Object next(){
                _____________________________________
                השלימו בדף התשובות
                _____________________________________
        }
}
```

<u>סעיף ג' (10 נקודות)</u>

השלם את השיטה getLevelsArray() במחלקה BinaryTree המחזירה מערך של מצביעים כך שהמצביע בתא i מצביע לרשימה של רמה i -- רשימה המכילה את נתוני הקודקודים (שדה ה-data) בעומק i (ראה איור 3). שים לב שאין חשיבות לסדר השדות.

הערה: ניתן להשתמש בשיטה height() במחלקה BinaryTree אשר מחשבת את גובהו של העץ(עומק מקסימאלי).



איור 3

```
public LinkedList[] getLevelsArray(){
        _____________________________________
        השלימו בדף התשובות
        _____________________________________

}
```

שאלה 3 (25 נקודות)

השלימו את הגדרת הפונקציה void subsetsSum(int[] aSet, int iK) אשר מקבלת מערך aSet של משקולות (ערכים שלמים חיוביים) ומשקל שלם iK חיובי (iK>0). הפונקציה תדפיס למסך את כל תתי הקבוצות של איברי המערך aSet שסכומם הינו iK.
לדוגמא, אם aSet = {1,2,3,4,5} ו- iK = 10 , הפונקציה תדפיס למסך:

1,2,3,4,
1,4,5,
2,3,5,

הניחו כי המערך aSet אינו null ושכל הערכים בו שלמים חיוביים, וכי iK>0. בנוסף, הניחו כי אין איבר המופיע במערך פעמיים. אין חשיבות לסדר האיברים בקבוצות המודפסות.

בתבנית הנתונה subsetsSum(int[]aSet, int iK) מתבצעת קריאה לפונקצית עזר **רקורסיבית** subsetsSum(...), בה ישנם מספר פרמטרים. השלימו את התבנית.

סעיף א' (5 נקודות)

```
public static void subsetsSum(int[] aSet, int iK)

        subsetSum(_______ השלימו בדף התשובות _______);

}
```

סעיף ב' (20 נקודות)

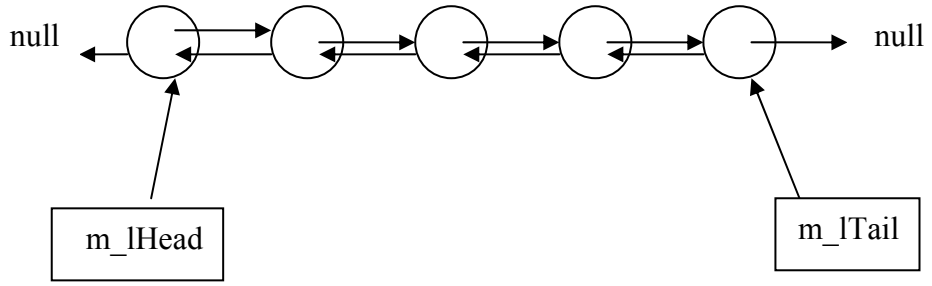
```
public static void subsetsSum ( _______ השלימו בדף התשובות _______){

        _______ השלימו בדף התשובות _______

}
```

שאלה 4 (25 נקודות)

רשימה מקושרת דו-כוונית (DoublyLinkedList) היא רשימה מקושרת שבה לכל איבר (DLink) יש גישה לאיבר הבא אחריו ברשימה ולאיבר הקודם לו ברשימה. בנוסף לשדות m_oData, וm_lNext , קיים שדה נוסף m_lPrevious, אשר מצביע לאיבר הקודם ברשימה:



שימו לב כי בנוסף למשתנה m_lHead , המצביע לתחילת הרשימה, אנו מחזיקים מצביע נוסף לסוף הרשימה m_lTail.

```
public class DLink{
        private DLink m_lNext;
        private DLink m_ lPrevious;
        private Object m_oData;

        public DLink(){
                m_lNext=null;
                m_oData=null;
                m_ lPrevious =null;
        }
        public DLink(Object oData){
                m_lNext=null;
                m_ lPrevious =null;
                m_oData= oData;
        }
        public DLink(DLink lNext,DLing lPrevious,Object oData){
                m_lNext= lNext;
                m_ lPrevious = lPrevious;
                m_oData= oData;
        }
        <acessors>
}//class

public class DoublyLinkedList{
        private DLink m_lHead;
        private DLink m_lTail;

         public DoublyLinkedList(){
                m_lHead=null;
                m_lTail=null;
        }
```

השיטה removeTail מסירה את האיבר האחרון ברשימה, מעדכנת את המצביע m_lTail, ומחזירה את המידע המשויך לאובייקט זה (oData).

```
public Object removeTail(){
        .
        ניתן להשתמש בשיטה זו
        .
}
```

השטיה ()size מחזירה את מספר האיברים ברשימה.

```
public int size(){
        .
        ניתן להשתמש בשיטה זו
        .
}
```

}//class DoublyLinkedList

בשלושת הסעיפים הבאים, עליכם לכתוב שיטות עבור המחלקה DoublyLinkedList.

<u>סעיף א'(5 נקודות)</u>

ממשו את השיטה addHead אשר מוסיפה אבר חדש לרשימה (המכיל את oData) בתחילתה.

```
public void addHead(Object oData){
        ____
        השלימו בדף התשובות
        ____
}
```

<u>סעיף ב'(5 נקודות)</u>

ממשו את השיטה removeHead אשר מסירה את האבר הראשון ברשימה, ומחזירה את המידע המשויך לאובייקט זה (oData). במידה והרשימה ריקה, השיטה תחזיר null.

```
public Object removeHead(){
        ____
        השלימו בדף התשובות
        ____
}
```

<u>סעיפים ג',ד'(15 נקודות)</u>
מחרוזת s הינה פלינדרום אם קריאתה משמאל לימין ומימין לשמאל הינה זהה. לדוגמא, המחרוזות "abcba", "abccba" הינם פלינדרום.
שימו לב: המחרוזת הריקה וגם מחרוזת בעלת תו אחד (לדוגמא "k", "") הינן פלינדרום.

בשאלה זו עליכם לממש את השיטה isPalindrome(String sExp) במחלקה PalindromeTest, המקבלת מחרוזת ובודקת האם היא פלינדרום באמצעות רשימה מקושרת דו כוונית.

השיטה תחילה מכניסה כל אחד מהתווים במחרוזת לרשימה. לאחר מכן קוראת לשיטת עזר אשר בודקת האם המחרוזת הינה פלינדרום.

הינכם רשאים להשתמש בשיטה substring(int i, int j) של המחלקה String אשר מחזירה את המחרוזת בין התו ה-i לתו ה-j:
לדוגמא :

```
String str="abcd";
System.out.println(str.substring(0,1)); → "a"
```

הערה: ניתן להניח כי המחרוזת sExp אינה null.

```
public class PalindromeTest{
```

<u>סעיף ג'(5 נקודות)</u>

```
    public static boolean isPalindrome (String sExp){
        DoublyLinkedList dllList=new DoublyLinkedList();

         השלימו בדף התשובות
        _____________________

        return isPalindrome (dllList);
    }
```

<u>סעיף ד'(10 נקודות)</u>

```
    private static boolean isPalindrome( DoublyLinkedList dllList){
         השלימו בדף התשובות
        ____________________

    }// isPalindrome

}//class
```

בהצלחה!

נספח

```
public class BinaryNode{
        private BinaryNode m_bnLeftChild, m_bnRightChild;
        private Object m_oData;
        public BinaryNode( Object oData ){..}

        public BinaryNode getLeftChild() {..}

        public BinaryNode getRightChild() {..}

        public void setLeftChild(BinaryNode left) {..}

        public void setRightChild(BinaryNode right) {..}

        public Object getData(){..}

        public void setData(Object dt){..}

}

public class BinaryTree{
        private BinaryNode m_bnRoot;

        public BinaryTree(){..}

        public int height(){..}
}

public class Link{
        private Link m_lNext;
        private Object m_oData;

        public Link(Object oData){..}

        public Link(Object oData,Link lNext){..}

        public void setNext(Link lNext){..}

        public Link getNext(){..}

        public Object getData(){..}

}
```

```
public class LinkedList {
        private Link m_lHead;
        public LinkedList(){..}
        public void addHead( Object oData ){..}

        public Object removeHead(){..}

        public boolean contains( Object oData ){..}

        public void addTail( Object oData ){..}

        public void remove(Object oData){..}

        public boolean isEmpty(){..}
}

public interface Comparable{
        public int compareTo( Comparable cOther );
}
```

מבוא למדעי המחשב 202-1-101-1

מבוא לתכנות למערכות מידע 202-1-104-1

סמסטר א', תשס"ז, 2006/7

בחינת מועד ב' – 16.2.2007

דף תשובות

שאלה 1 (20 נקודות)

סעיף א' (5 נקודות)

```
public boolean isEmpty(){
```

_____

```
}
```

סעיף ב' (15 נקודות)

```
public void insert(Comparable cData){
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
}
```

שאלה 2 (30 נקודות)

סעיף א'(10 נקודות)

```
private void computeLevelList(LinkedList llLevel, int iDepth, BinaryNode bnNode){
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
}
```

סעיף ב'(10 נקודות)

```
public class LevelIterator implements Iterator{
        _____________________________________________
        _____________________________________________
        _____________________________________________
        _____________________________________________
        public  LevelIterator( _____________________________________){
                _____________________________________________
                _____________________________________________
                _____________________________________________
                _____________________________________________
                _____________________________________________
                _____________________________________________
        }
        public boolean hasNext(){
                _____________________________________________
                _____________________________________________
        }
```

```
        public Object next(){
                _____
                _____
                _____
                _____
                _____
                _____
        }
}
```

<u>סעיף ג'(10 נקודות)</u>

```
public LinkedList[] getLevelsArray(){
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
        _____
}
```

שאלה 3 (25 נקודות)

סעיף א' (5 נקודות)

```
public static void subsetsSum(int[] aSet, int iK)

        subsetSum(_____________________________________________);

}
```

סעיף ב' (20 נקודות)

```
 public static void subsetsSum (_____________________________________________){
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
        _____________________________________________________________
}
```

שאלה 4 (25 נקודות)

סעיף א'(5 נקודות)

```
public void addHead(Object oData){
```

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

```
}
```

סעיף ב'(5 נקודות)

```
public Object removeHead(){
```

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

```
}
```

<u>סעיף ג'(5 נקודות)</u>

```
public static boolean isPalindrome (String sExp){
        DoublyLinkedList dllList=new DoublyLinkedList();
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
        return isPalindrome (dllList);
}
```

<u>סעיף ד'(10 נקודות)</u>

```
public static boolean isPalindrome( DoubleLinkedList dllList){
                ____________________________________________________
 _
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________
            _____________________________________________________


}// isPalindrome
```

בהצלחה!