

## מעבדה 6. נושא: עץ חיפוש בינארי

תאריך הגשה לקבוצה של יום ב': 07.12.2020

תאריך הגשה לקבוצה של יום ד': 09.12.2020

(הגשה בזוגות)

**יש לקרוא היטב לפני תחילת העבודה!**

עבודה נעימה!

### מבוא:

במעבדה הנוכחית נממש עצי חיפוש בינאריים.

### תיאור:

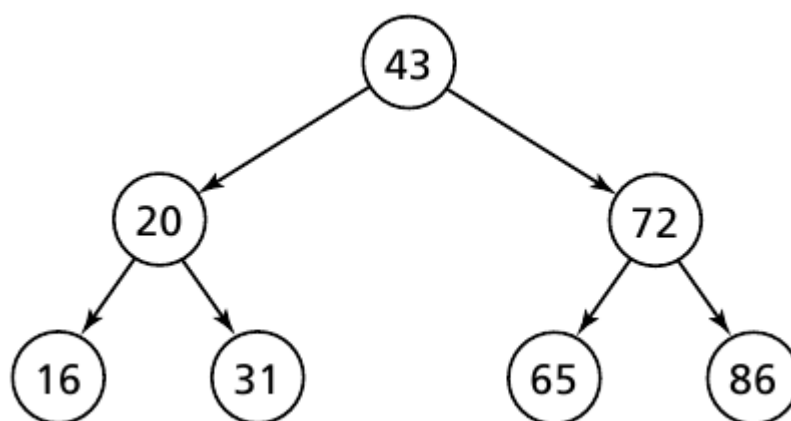
#### הגדרה 1 (עץ בינארי):

עץ שלכל אחד מהצמתים הפנימיים שלו יש לכל היותר שני בנים, נקרא עץ בינארי

#### הגדרה 2 (עץ חיפוש בינארי):

עץ בינארי בו מפתחות של כל הצמתים הנמצאים בתת-עץ שמאלי קטנים ממפתח של שורש העץ ומפתחות של כל הצמתים הנמצאים בתת-עץ ימני גדולים ממפתח של שורש העץ, נקרא עץ חיפוש בינארי.

דוגמא לעץ חיפוש בינארי:



נממש את עץ החיפוש לפי השלבים הבאים:

- בשלב ראשון ניצור מחלקה המכילה **צומת** בודד של העץ.
- בשלב שני **ניצור מחלקה עבור עץ חיפוש בינארי**.

## שלב 1.

ניצור מחלקה של **צומת בעץ חיפוש** (BSTreeNode). כל איבר יכיל מפתח (**key**) ושני מצביעים. להלן המחלקה:

```
public class BSTreeNode {
    private int key;
    private BSTreeNode left, right;
    public BSTreeNode ( int key, BSTreeNode leftPtr, BSTreeNode rightPtr ){ }

    // ---Insert set/get methods here ---//
}
```

צור קובץ בשם BSTreeNode.java, וממש את כל המטודות של המחלקה לעיל.

## שלב 2.

צור מחלקה המגדירה **עץ חיפוש בינארי** (BSTree). שים לב: עליך להוסיף שיטות **רקורסיביות** **private methods** הדרושות למימוש העץ.

(תזכורת: הוספה של צומת מתבצעת רק אם הצומת הוא **עלה!**)

להלן המחלקה:

```
public class BSTree{
    private BSTreeNode root;

    public BSTree ( ){ }

    // Binary search tree manipulation methods
    public void insert ( int newKey ){ }
    public BSTreeNode retrieve ( int searchKey ){ }
    public void clear ( ){ } // Clear tree

    // Binary search tree status methods
    public boolean isEmpty ( ){ }
    public boolean isFull ( ){ }

    // Print tree methods
    public String Preorder(){ }
    public String Inorder(){ }
    public String Postorder(){ }
```

**// Recursive partners of the public member methods --- Insert these methods here.**

```
} // class BSTree
```

צור קובץ בשם BSTree.java, וממש את המחלקה המופיעה לעיל.

### שלב 3. בדיקות

צור קלאס חדש בשם `TestBSTree.java` ובדוק בעזרתו את נכונות המחלקות שכתבת. על המחלקה לתמוך בפעולות הבאות:

Command	Action
+key	Insert (or update) the element with the specified key.
?key	Retrieve the element with the specified key and output it.
-	-----
K	Output the keys in ascending order.
E	Report whether the tree is empty.
F	Report whether the tree is full.
C	Clear the tree.
Q	Quit the test program.

**הערה: במקום ?+ אפשר להשתמש ב `add, find`**

### שלב 4. פעולות נוספות

נגדיר **גובה העץ** = מספר קשתות במסלול הארוך ביותר משורש העץ עד לעלה כלשהו.

- למחלקה `BSTree` הוסף מטודה המחשבת את גובה העץ.
- כמו כן הוסף למחלקה `TestBSTree` פקודה `H` הבודקת את נכונות כתיבת המטודה הנ"ל.

**`int height ()`**

Precondition: None.

Postcondition: Returns the height of a binary search tree.

### הגשה:

`BSTree.java`, `BSTreeNode.java`, `TestBSTree.java`

### **ערעורים:**

יש להפנות לד"ר תמר צמח בלבד ע"י שליחת מייל [tamar.zemach@yahoo.com](mailto:tamar.zemach@yahoo.com) **תוך שבוע** ממועד פרסום הציונים. בכותרת המייל יש לציין: "ערעור מעבדה מס' X ע"י +ת.ז. של הסטודנטים". יש לקחת בחשבון שבעת הערעור העבודה נבדקת מחדש וכתוצאה מהערעור ציון העבודה עלול להשתנות (יכול לעלות או לרדת). התשובה לערעור תהיה סופית ולא ניתנת לערעור נוסף.