שאלות

שאלה 1

הוכח או הפרך: אם X הינו $Node$ בעץ חיפוש בינארי אשר ישלו 2 בנים לא ריקים אזי ל$Successor$ שלו אין בן שמאלי.

שאלה 2

כתבו פונקציה המקבלת שורש של עץ ($Node\ a$) ומחזירה $True$ אם העץ הינו מאוזן , כלומר ה $Balance\ Factor$ לכל קודקוד הינו בין 1 ל -1 (בשאלה זו למחקלת $Node$ **אין** שדה של גובה $Height$).

שאלה 3

ממשו את הפונקציה של $Successor$.

חתימת הפונקציה: $public\ Node\ Successor(Node\ n)$

שאלה 4

ממשו את הפונקציה של $Predecesor$.

חתימת הפונקציה: $public\ Node\ Predecesor(Node\ n)$

שאלה 5

נכון או לא נכון נמקו היטב :

תת העץ השמאלי של שורש של עץ AVL תמיד מקיים את כל התכונות של עץ AVL.

שאלה 6

צייר עץ AVL לאחר הוספה של כל אחד מהאיברים הבאים (משמאל לימין) : $5, 6, 7, 1, 2, 3$.

ס"הכ 6 ציורים.

שאלה 7

הוכח או הפרך: הכנסה בעץ AVL עם n קודקודים תיקח $\theta(n)$ רוטציות.

פתרונות

פתרון שאלה 1

הוכחה: כיוון שיש ל X שני בנים לא רקים זה אומר שיש לו גם בן ימני ולכן לפי הגדרה ה $Successor$ של X הינו הבן השמאלי ביותר בתת העץ הימני שלו. מכיוון שהוא השמאלי ביותר בתת עץ זה(כלומר המינאמלי) אין איבר שקטן ממנו ולכן אין ל$Successor$ בן שמאלי.

פתרון שאלה 2

```
1  public static boolean isBalanced(Node root) {
2    if(root == null)
3      return true;
4    else
5      return (isBalanced(root.right) && isBalanced(root.left) &&
6      (Math.abs(height(root.right) - height(root.left)) <= 1);
7  }
8  public static int height(Node root) {
9    if(root == null)
10     return -1;
11   else
12     return Math.max(height(root.left),height(root.right)) + 1;
13 }
```

פתרון שאלה 3

```
public Node Successor(Node n){
    // first case if has right subtree
    if (n.right != null) {
        return minValue(n.right);
    }

    // second case
    Node p = n.parent;
    while (p != null && n == p.right) {
        n = p;
        p = p.parent;
    }
    return p;
}
//simply find left most leaf
public Node minValue(Node node){
    Node current = node;

    /* loop down to find the leftmost leaf */
    while (current.left != null) {
        current = current.left;
    }
    return current;
}
```

פתרון שאלה 4

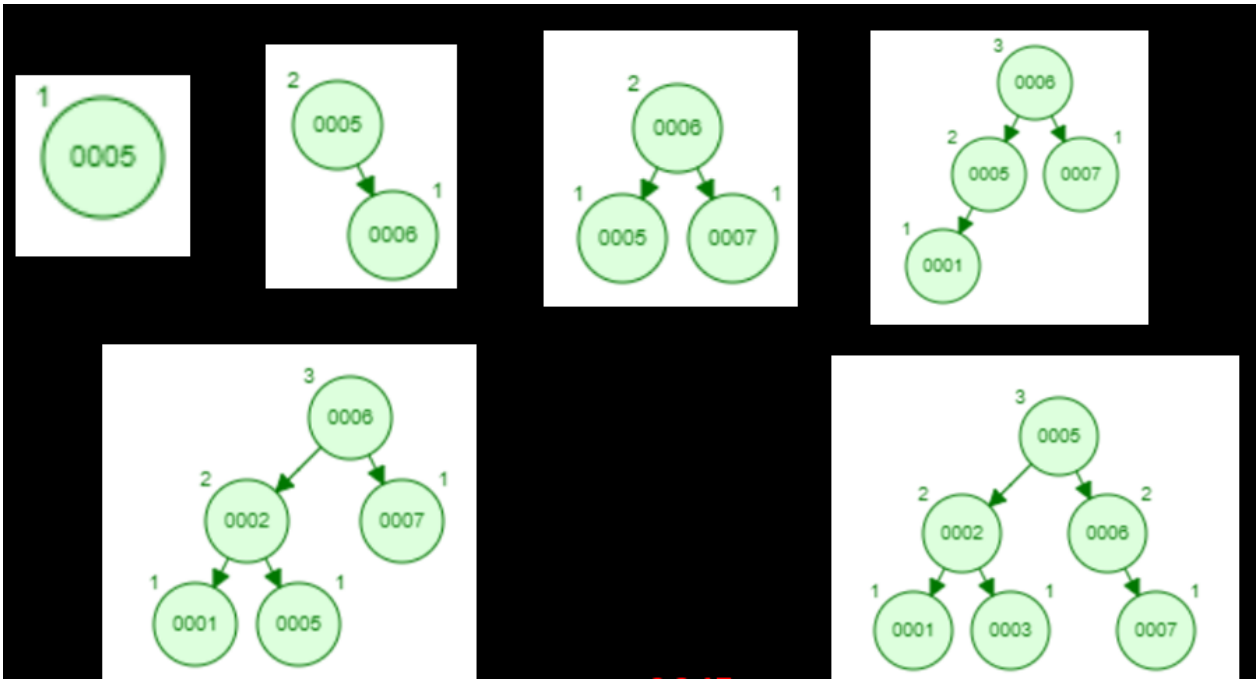
```
public Node Predecesor(Node n){
    // first case if has left subtree
    if (n.left != null) {
        return maxValue(n.left);
    }

    // second case
    Node p = n.parent;
    while (p != null && n == p.left) {
        n = p;
        p = p.parent;
    }
    return p;
}
//simply find right most leaf
public Node minValue(Node node){
    Node current = node;

    /* loop down to find the rightmost leaf */
    while (current.right != null) {
        current = current.right;
    }
    return current;
}
```

פתרון שאלה 5

תשובה: נכון

הסבר: ניתן להסביר זאת ריקורסבית, לפי הגדרה עץ avl מקיים שלכל צומת בעץ הצומת מאוזן. כלומר כל צומת מקיים $|height(node.right) - height(node.left)| \leq 1$. לכן מה שאנו רוצים להוכיח הוא שאם ניקח את תת העץ השמאלי הוא גם ייקים את התכונה הזאת של avl (זה ברור שהוא בינארי). מכיוון שכל צומת של העץ הגדול מקיימת תכונה זו של האיזון אז בפרט מתקיים שגם לתת עץ השמאלי של השורש הוא מאוזן וזאת מכיוון שאם אני הולך לתת עץ הימני והשמאלי שלו הצמתים שם גם מאוזנים וכן הלאה.

פתרון שאלה 6



פתרון שאלה 7

הפרכה , כל הכנסה גוררת או 0 רוטציות או רוטציה אחת או 2 רוטציות. לכן סך כמות הרוטציות הוא $\theta(1)$