**King Fahd University of Petroleum & Minerals**

**College of Computer Science and Engineering**

**Information and Computer Science Department**

**ICS 202 – Data Structures**

# AVL Trees

**Objectives**

The objective of this lab is to design, implement and use AVL trees.

**Outcomes**

After completing this Lab, students are expected to:

• Design classes for AVL trees.

• Know Insertion and Deletion cases of AVL trees.

**Notes**

The given example in the test program AVL\_Driver.java verifies the following AVL insertions that require a single left rotation when 45 is inserted:

Diagram

Description automatically generated

Execute the given test program to get the following output:

SINGLE LEFT ROTATION EXAMPLE:   
Insert 45 in the following AVL tree:  
Root--30  
 L----5  
 R----35  
 L----32  
 R----40  
  
Insertion result:   
Balance factor = 2  
Balancing node with el: 30  
LEFT ROTATION  
  
Root--35  
 L----30  
 | L----5  
 | R----32  
 R----40  
 R----45

**Lab Tasks**

**Note**: The given **AVL<T extends Comparable<T>> extends BST<T>** class is incomplete, specifically the methods rotateRight(), rotateRightLeft() and deleteAVL(T el) have not been implemented. You are required to complete these methods and make necessary additions to the test program in order to do the lab tasks.

1. Verify the following insertion by the test program:

Diagram, schematic

Description automatically generated

to get the output given below:

DOUBLE RIGHT-LEFT ROTATION EXAMPLE:   
Insert 15 in the following AVL tree:  
Root--5  
 L----2  
 | L----1  
 | R----4  
 | L----3  
 R----7  
 L----6  
 R----9  
 R----16  
  
Insertion result:   
Balance factor = 2  
Balancing node with el: 9  
Double Rotation...  
RIGHT ROTATION  
LEFT ROTATION  
  
Root--5  
 L----2  
 | L----1  
 | R----4  
 | L----3  
 R----7  
 L----6  
 R----15  
 L----9  
 R----16

1. Verify the following deletion by the test program:

Diagram, schematic

Description automatically generated

to get the output given below:

CASE 3A DELETION EXAMPLE:   
Delete 1 in the following AVL tree:  
Root--7  
 L----2  
 | L----1  
 | R----3  
 | R----5  
 R----15  
 L----10  
 | L----9  
 | R----13  
 | L----11  
 R----17  
 R----18  
  
Deletion result:   
Balance factor = 2  
Balancing node with el: 2  
LEFT ROTATION  
Balance factor = 2  
Balancing node with el: 7  
Double Rotation...  
RIGHT ROTATION  
LEFT ROTATION  
  
Root--10  
 L----7  
 | L----3  
 | | L----2  
 | | R----5  
 | R----9  
 R----15  
 L----13  
 | L----11  
 R----17  
 R----18

1. What is the maximum integer key that when inserted in the AVL tree below will cause a single right rotation?

Verify your answer by the test program.



The output must be in the form shown below where the insertion part is generated by your program:

MTASK03:   
Insert the required maximum key in the following AVL tree:  
Root--32  
 L----26  
 | L----14  
 | R----30  
 | L----27  
 R----54  
 L----44  
  
Insertion result:

1. What is the minimum integer key that when inserted in the AVL tree below will cause a double left-right rotation? Verify your answer by the test program.



The output must be in the form shown below where the insertion part is generated by your program:

TASK04:   
Insert the required minimum key in the following AVL tree:  
Root--32  
 L----26  
 | L----14  
 | R----30  
 | L----27  
 R----54  
 L----44  
  
Insertion result: