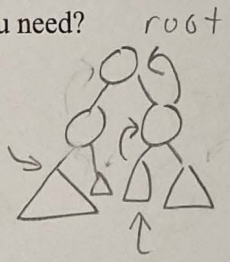
EECS 233 SI Session 8
Leader: Bertram Su
October 3, 2019

**Objectives:**
Upon completion of this SI session, participants will be able to:

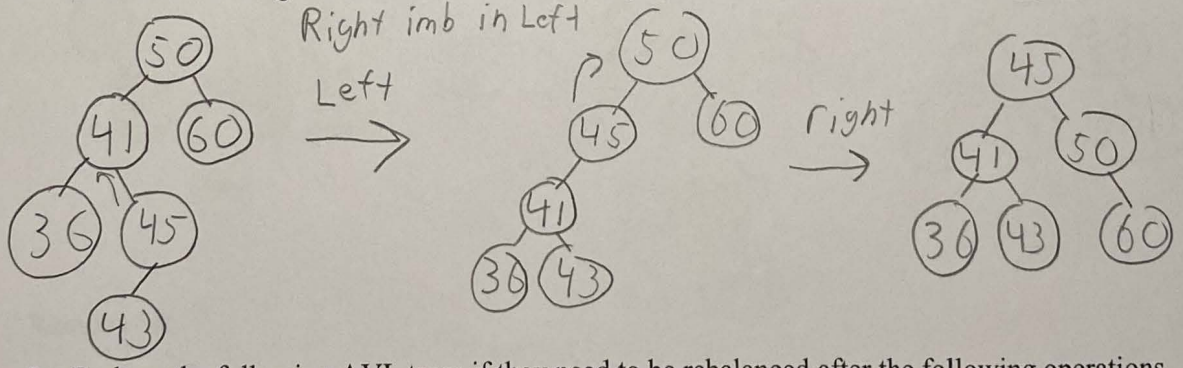
1. Recognize what add and remove would do to an AVL tree
2. Implement double rotations in code

**Exercises:**

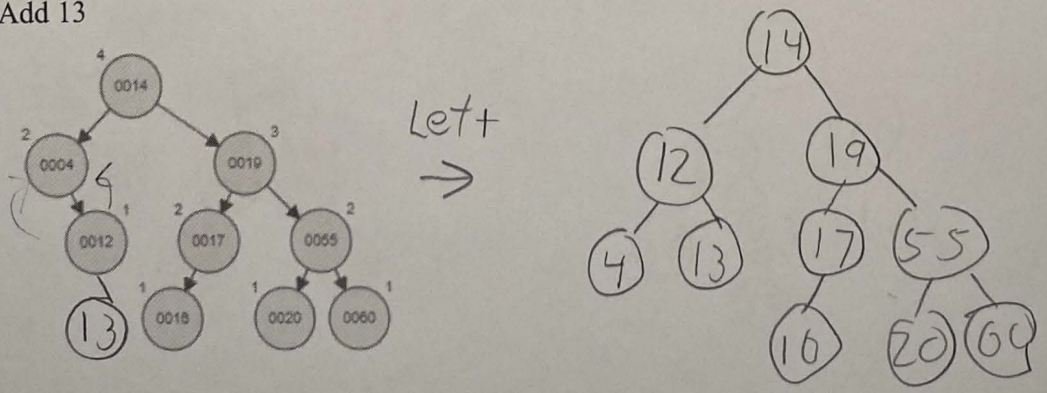
1. Given the following situations what type of rotation(s) would you need?
   a) A right imbalance in the left subtree
      Left right
   b) A right imbalance in the right subtree
      Left
   c) A left imbalance in the left subtree
      right
   d) A left imbalance in the right subtree
      right Left



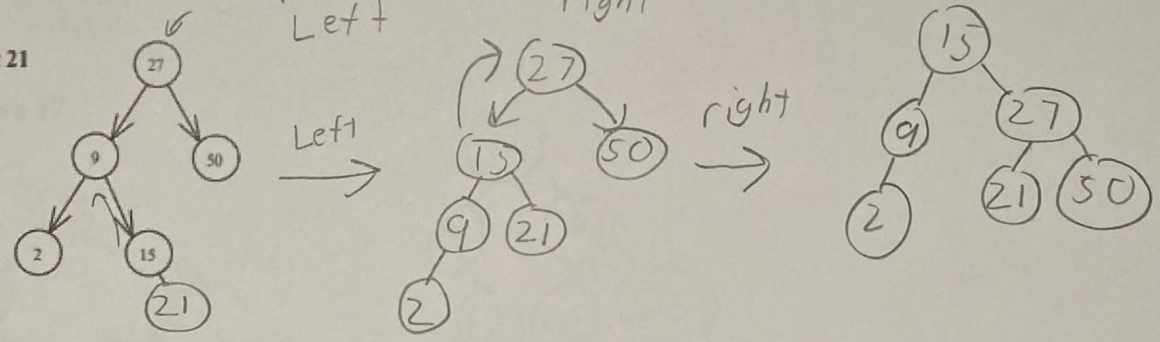

2. Draw the resulting AVL tree after the following operations: Add 50, 60, 41, 45, 36, 43



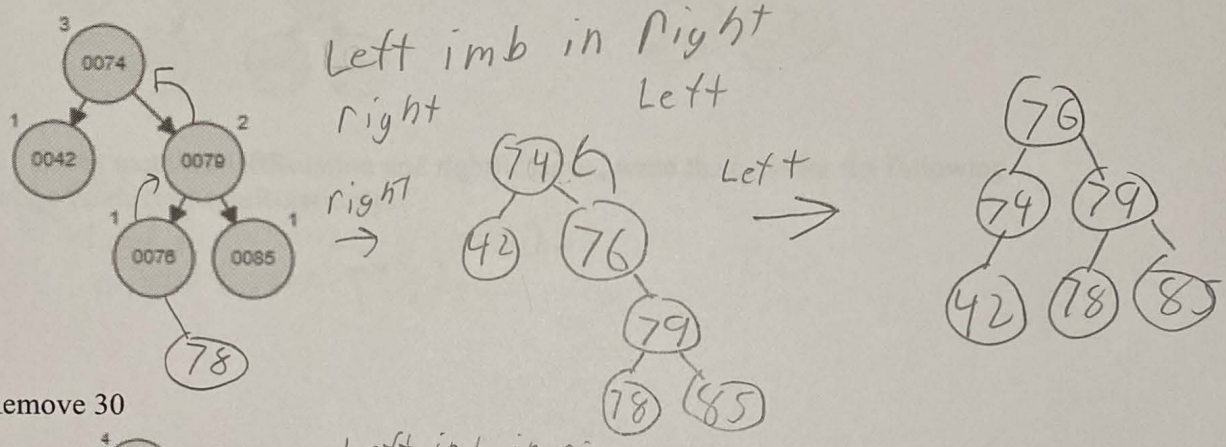

3. Redraw the following AVL trees if they need to be rebalanced after the following operations
   Add 13



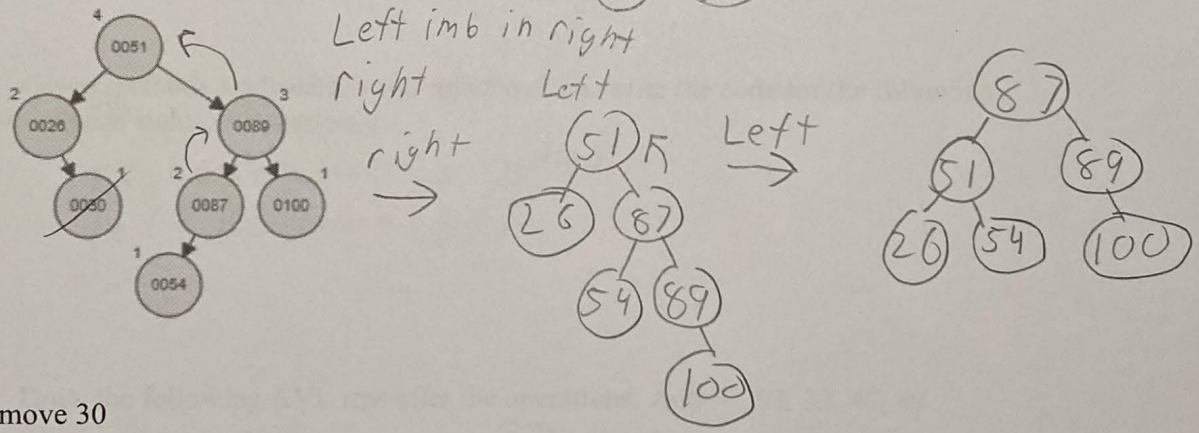
**Insert 21**



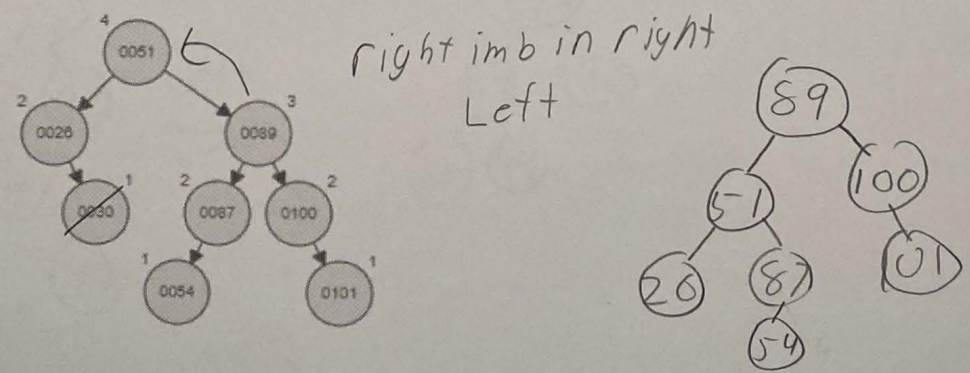

Add 78



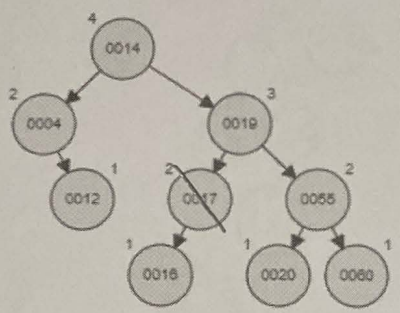

Remove 30

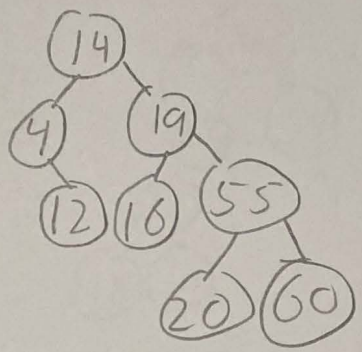



Remove 30



Remove 17




4. Given methods leftRotation and rightRotation, write the code for the following
public ~~Node~~ void leftRightRotation(y){

```
leftRotation(y.left);
rightRotation(y);
```

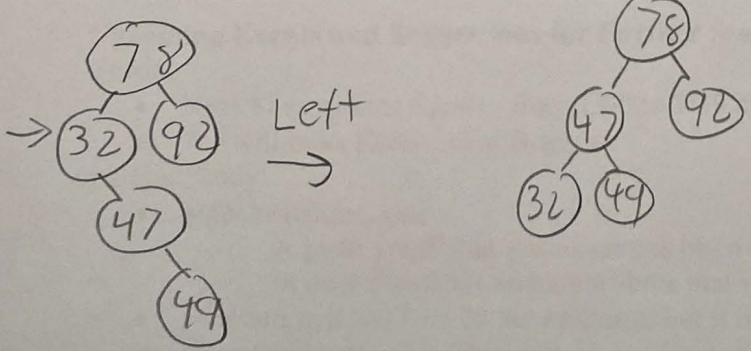
}

5. Given methods leftRotation and rightRotation, write the code for the following
public ~~Node~~ void rightLeftRotation(y){

```
rightRotation(y.right);
leftRotation(y);
```

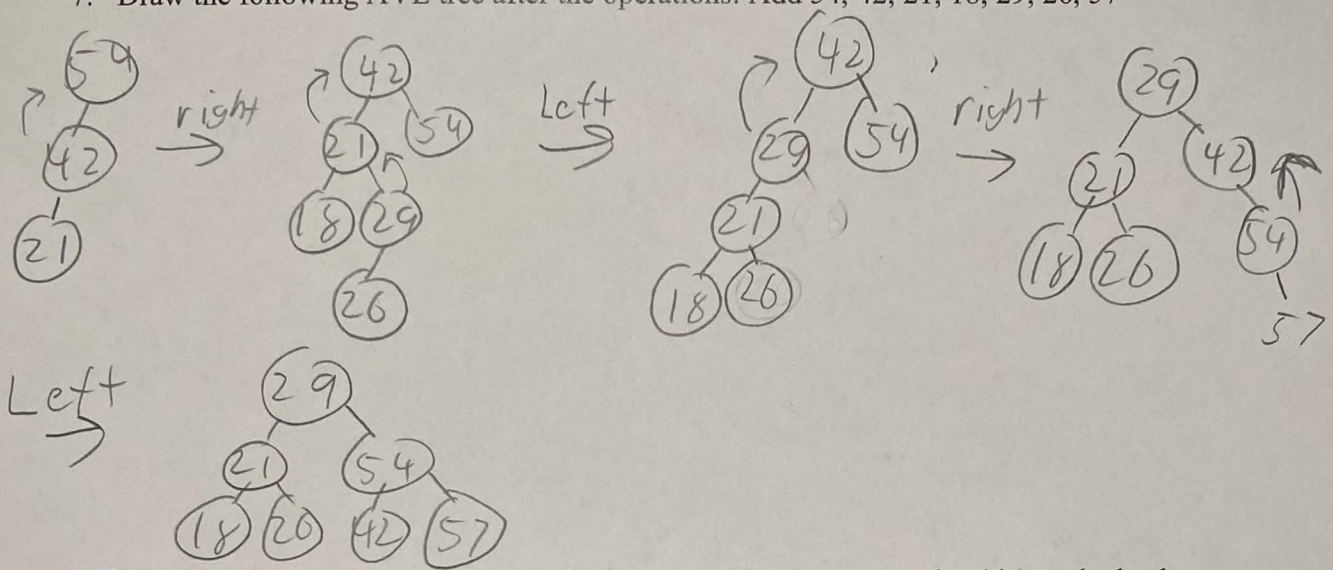
}

6. Draw the following AVL tree after the operations: Add 78, 92, 32, 47, 49



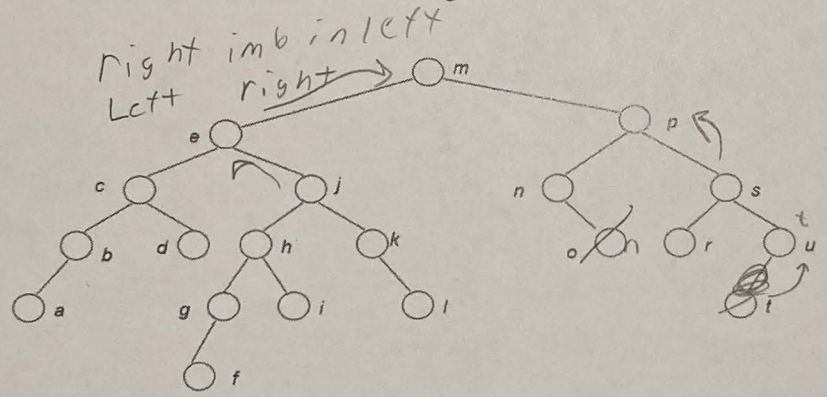
**Summary**

7. Draw the following AVL tree after the operations: Add 54, 42, 21, 18, 29, 26, 57




8. If we have time* (I would recommend using a lined piece of paper for this) or the back

***Exercise:*** Delete *o* from the following AVL tree




**Upcoming Events and Suggestions for Further Study:**

Events:

- Next SI session is Sunday from 1:00 to 2:30 at Sears 336
- We will most likely cover B-trees

Further Study:

- bigocheatsheet.com
  - A great graph that visualizes the big o complexity chart. It also has the big O time of data structures and algorithms that we will cover in the future.
- Deletion will NOT be on the midterm, but it may show up in the homework.
- https://www.cs.usfca.edu/~galles/visualization/AVLtree.html
  - Practice AVL trees

EECS 233 SI Session 9
Leader: Bertram Su
October 5, 2019

**Objectives:**
Upon completion of this SI session, participants will be able to:

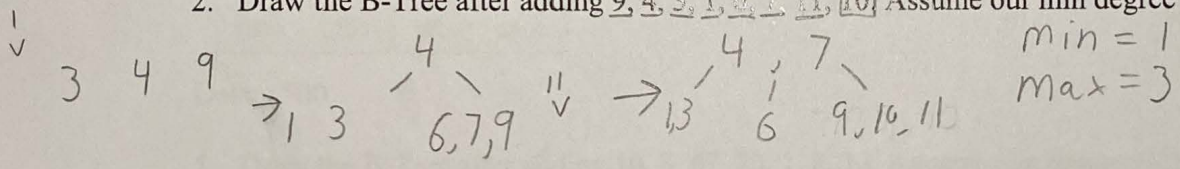
1. Recognize B-tree terminology
2. Determine what adding would do to a B-Tree of varying degrees

**Foundations:**

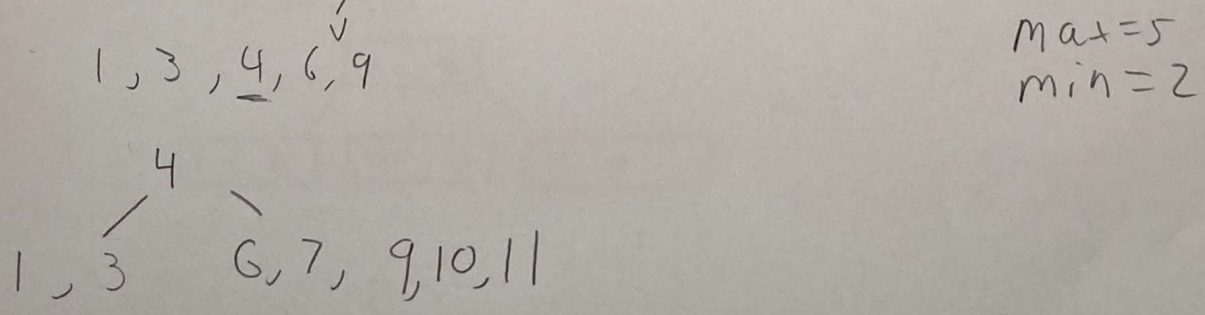
1. Given a minimum degree t
   a) What's the minimum amount of keys each non-root node must have?
      t-1
   b) What's the maximum amount of keys each node can contain?
      2t-1
   c) What's the minimum amount of keys a root must have?
      1

**Exercises:**

2. Draw the B-Tree after adding 9, 4, 3, 1, 6, 7, 11, 10. Assume our min degree is 2



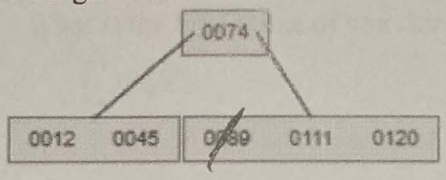

3. Draw the B-Tree after adding 9, 4, 3, 1, 6, 7, 11, 10. Assume our min degree is 3



4. What are some possible B-Trees after the following operations?

Min degree = 3

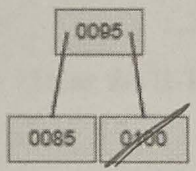
Max = 5
min = 2




Delete 89

Min Degree = 2 (There are two possible solutions)

min = 1
max = 3

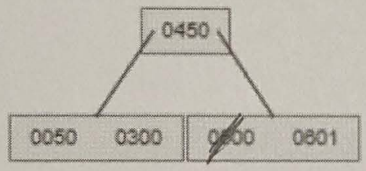



85, 95

Delete 100

Min Degree = 3

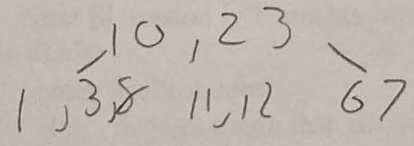
Max = 5
min = 2



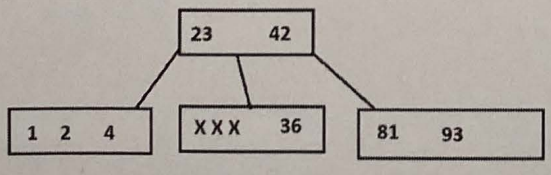

50, 300, 450, 601

Delete 500

5. Draw the B-Tree after adding 10, 3, 67, 23, 1, 8, 12, 11. Assume min degree is 2




6. What is the possible range of values for the three variables X? The min 2 and assume no repeated integers




$23 < X < 36$

**Summary**

7. What is the big O time of searching for something in a B-Tree? $t << n$

   $\log n$

8. When would we use B-Trees?

   No rotations, going through memory

   fewer random jumps

   degree

9. Draw the B-Tree after adding 10, 3, 67, 23, 1, 8, 12. Assume min is 3

   8

   1, 3, 10, 23, 67 →

   10

   1, 3, 8 12, 23, 67

   min = 2

   max = 5

**Upcoming Events and Suggestions for Further Study:**

Events:

- Next SI session is Thursday from 1:00 to 2:30 at Sears 336

Further Study:

- bigocheatsheet.com
  - A great graph that visualizes the big o complexity chart. It also has the big O time of data structures and algorithms that we will cover in the future
- https://www.cpp.edu/~ftang/courses/CS241/notes/b-tree.htm
  - This website has examples of addition and removal