**Class Activity 34**

1. Binary Search Tree:
   1. What is the Big O time complexity and why? (**Hint: worst-case and best-case**)

**Worst-case:** O(n) because every element could be traversed

**Best-case:** is O(Log n) because typically most of time, half the tree can be skipped after each look-up

Min Heap:

* 1. What is the Big O time complexity and why? (**Hint: worst-case and best-case**)

**Worst-case**: O(log n) because min needs to be maintained at its root node which takes Log n operations.

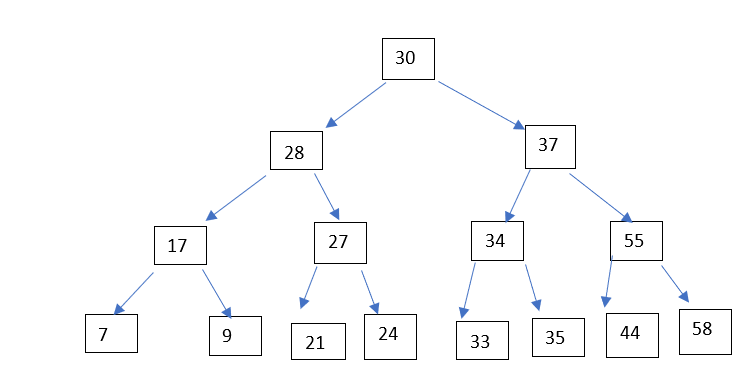
**Best-case**: O(1) because it starts with the minimum value to begin with

1. Max Heap:
   1. What is the Big O time complexity and why? (**Hint: worst-case and best-case**)

Worst-case: O(log n) because it would be a single element comparisons multiplied by n

Best-case: O(1) same reason as above, except starting at the max value

1. In a Binary tree, each node has up to **2** children.
2. In an n-nary tree, each node has up to **n** children.
3. A node in a binary tree that has no children is called what? **Leaf**
4. A **min heap** is a complete binary tree where each node is smaller than or equal to its children, commonly used in priority queues.
5. Use the following tree structure to perform the listed operations.



**Identify the**:

1. Root node: 30
2. Three branch nodes:17, 27, 34
3. Three Leaf nodes: 7, 9, 21
4. Depth of tree: 3
5. Size of Tree: 15
6. Height: **3**

Preorder Traversal **(Root , Left, Right**): 30, 28, 17, 7, 9, 27, 21, 24, 37, 34, 33, 35, 55, 44, 58

Inorder Traversal (**Left, Root, Right**): 7, 17, 9, 28, 21, 27, 24, 30, 33, 34, 35, 37, 44, 55, 58

Postorder Traversal (**Left, Right, Root**): 7, 9, 17, 21, 24, 27, 33, 35, 34, 44, 58, 55, 28, 37, 30