Topics:

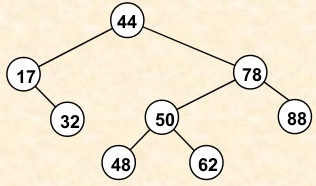
1. The efficiency of algorithms:
   1. Concept of Big O
2. Binary search trees
   1. Concept (for example, the keys of left subtree is always less than the keys of the right subtree)
   2. Create a tree (by inserting a sequence of numbers)
   3. Search
   4. Insert
   5. Delete
3. AVL trees
   1. Concept (for example, rotation)
   2. Create a tree (by inserting a sequence of numbers)
   3. Search
   4. Insert
   5. Delete (You should know the main concept. If there is a question, I will provide the algorithm and you must able to apply the deletion algorithm for a sample of numbers)
4. Red-black tree
   1. Concept (like rotation and recoloring)
   2. Create a tree (by inserting a sequence of numbers)
   3. Search
   4. Insert ()
   5. Delete (You should know the main concept. If there is a question, I will provide the algorithm, and you have to able to apply the deletion algorithm in an example)
5. M-way tree, B-tree/B+-tree
   1. Concept
   2. Create a tree
   3. Search
   4. Insert (may need to use split operation)
   5. Delete (The algorithm will be provided)

Practice Questions:

1. Insertion order (Write step-by-step insertion with all the necessary operations):

70, 20, 60, 30, 50, 65

1. Binary search tree
2. AVL Tree
3. Red-black Tree
4. Traverse the following tree



1. Pre-order:
2. In-order:
3. Post-order:
4. Breath-first:
5. Consider the AVL deletion (node X) algorithm:

Step 1) Deletion:

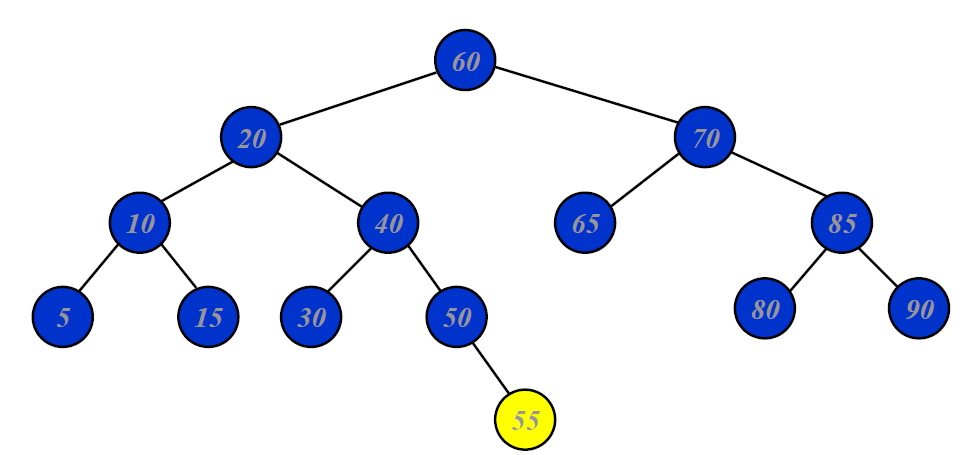
■Case 1: if X is a leaf, delete X

■Case 2: if X has 1 child, use it to replace X

■Case 3: if X has 2 children, replace X with its inorder predecessor(and recursively delete it)

Step2) Rebalancing (proper rotation to keep the tree balanced)

Delete the nodes from the following tree:

1. Delete 55
2. Delete 50 (without considering deletion of 55)
3. Delete 60 (without considering deletion of 55 and 50)