1. Set

Example:

In list

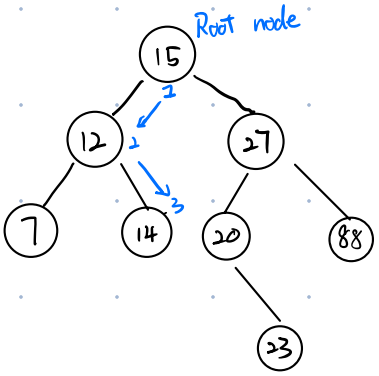
["USA", "India", "China", "India", "USA"]

In Set

["USA", "India", "China"]

Explanation: Set will remove all the duplicates

2. Binary Search Tree



(1) Search for number 14

(1) The Binary Search Tree has at most two child nodes

(2) Order:

1. All the elements on the left-hand side have value less than parent node.

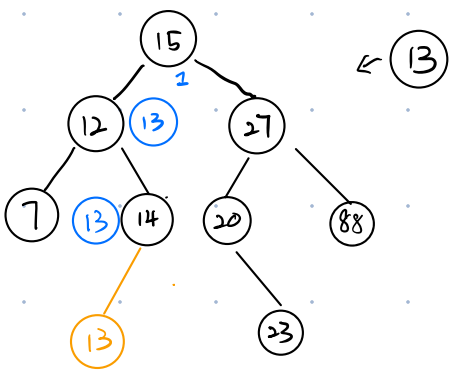
2. All the elements on the right-hand side have value more than parent node.

(3) Search Complexity

O(log n)

Because every iteration we reduce search space by 1/2

(2) Insert element



Steps:

1. Compare to each nodes

(1) Compare to 15, less so go to left way

(2) compare to 12, more so go to right way

(3) compare to 14, less so go to left way

- Breadth-first Search

(1) We use queue to store and pop out the data

(2) Visiting all nodes at each level before moving out to the next

- Depth-first Search

(1) We use stack (or recursion)

(2) explores as far down one branch as possible before backtracking to explore other branch.

Three main types of traversal methods: In-order traversal, Pre-order traversal
Post-order traversal

1. In-order traversal

order: left subtree → root node → right subtree

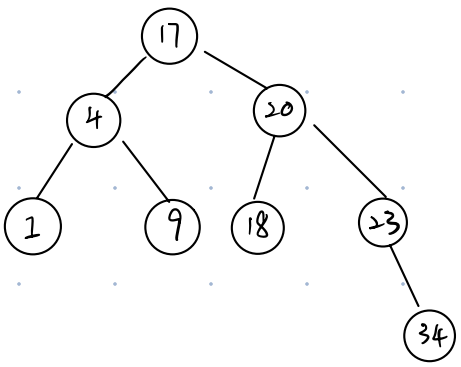
2. Pre-order traversal

order: root → left subtree → right subtree

3. Post-order traversal

order: left subtree → right subtree → root

- Delete the node from a binary search



(1) Deleting a leaf node

9

(2) Deleting node with one child

23

(3) Deleting node with two child

20

(1) You can pick right or left subtree to start delete the point node.

(2) copy the minimum value in the right subtree
copy the maximum value in the left subtree