BST

✓Varient of Binary tree

✓

Every node must have atmost(0, 1, 2) two children

Binary Tree + Binary Search = Binary Search Tree

All the elements of left sub tree are less than Root.

All the elements of right sub tree are Greater than Root.

Delete

1. Deleting a node with no child
2. Deleting a node having one child
3. Deleting a node having two child

Deleting a node with no child

Deleting a node having one child

Child will replace the position of parent

Deleting a node having two child

Replace parent with Child

Child = Minimum element of Right Sub tree

Child = Maximum element of Left Sub tree

DISJOINT SETS

Definition: If S_i and S_j are two sets and $i \neq j$ and there is no element that is in both S_i and S_j, then the two sets S_i and S_j are called disjoint sets. For example, consider the following three sets:

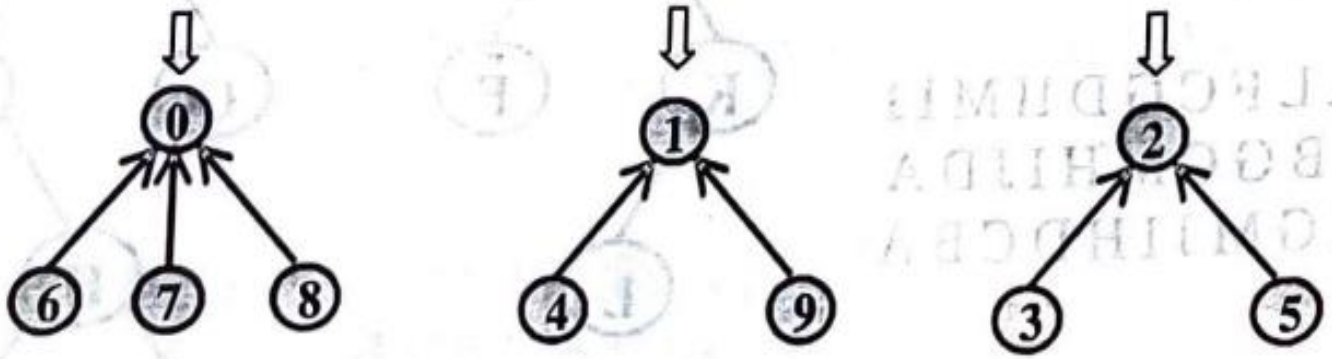
$$S_0 = \{0, 6, 7, 8\} \quad S_1 = \{1, 4, 9\} \quad S_2 = \{2, 3, 5\}$$

Note that there are no common items in any of the three sets S_1, S_2 and S_3. So, the sets S_1, S_2 and S_3 are disjoint sets.

Cont..

For example, consider three disjoint sets and their tree representations

$S_0 = \{0, 6, 7, 8\}$ $S_1 = \{1, 4, 9\}$ $S_2 = \{2, 3, 5\}$



Note that the first item in each set is the subscript in the name of the set and it will be the root node. For example, in set $S_0 = \{0, 6, 7, 8\}$, the name of the set is S_0. The subscript 0 in S_0 indicates that 0 is the first element of set S_0. All other remaining items are the children of 0 from left to right and the directions are given from children to the parent.

Since set is collection of similar data items, the set can be implemented using arrays. The sets S_0, S_1 and S_2 represented in the form of a tree can be implemented using array as shown below:

- When a node does not have a parent, let us store -1 in the array using appropriate index. The root nodes in all the three trees i.e., node 0 in first tree, node 1 in second tree and node 2 in third three do not have parents. So, in the array, we can write:

$$S[0] = S[1] = S[2] = -1$$

- The parent for node 6, 7, 8 is the node 0. So, in the array, we can write:

$$S[6] = S[7] = S[8] = 0$$

- The parent for node 4 and node 9 is node 1. So, in the array, we can write:

$$S[4] = S[9] = 1$$

- The parent for node 3 and node 5 is 2. So, in the array, we can write:

$$S[3] = S[5] = 2$$

So, the array corresponding to sets S_0, S_1 and S_2 is shown below:

i	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
S	-1	-1	-1	2	1	2	0	0	0	1

Operations on Disjoint sets

- ✓Union
- ✓Find

union

6.15.1 Disjoint set union

If S_i and S_j are two disjoint sets, then their union is given by:

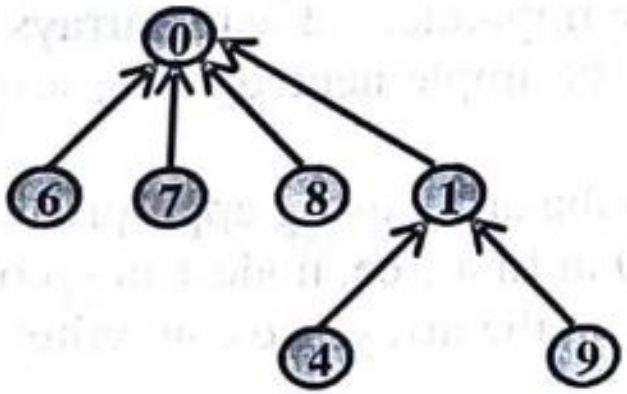
$$S_i \cup S_j = \{ \text{ all elements } x \mid x \text{ is in } S_i \text{ or } S_j \}$$

For example, let $S_0 = \{0, 6, 7, 8\}$ and $S_1 = \{1, 4, 9\}$ then,

$$S_0 \cup S_1 = \{0, 1, 4, 6, 7, 8, 9\}$$

After finding $S_0 \cup S_1$, the sets S_0 and S_1 does not exist where as their union $S_0 \cup S_1$ exists. That is, S_0 and S_1 will be replaced by their union $S_0 \cup S_1$. The tree equivalent of $S_0 \cup S_1$ is shown below:

Equivalent tree & Array representation

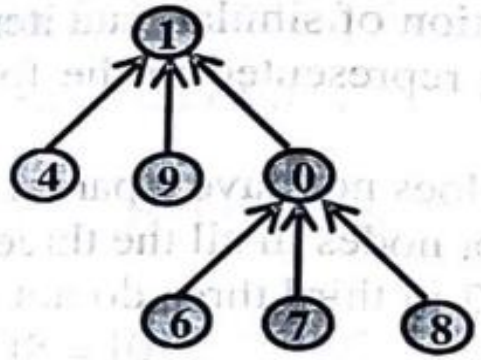


$S_0 \cup S_1$

Note: The root node corresponding to set S_1 will be the child for root node 0. The array representation is shown below:

s[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
-1	0			1		0	0	0	1

Equivalent tree & Array representation



$S_0 \cup S_1$

Note: The root node corresponding to set S_0 will be the child for root node 1. The array representation is shown below:

s[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	-1			1		0	0	0	1

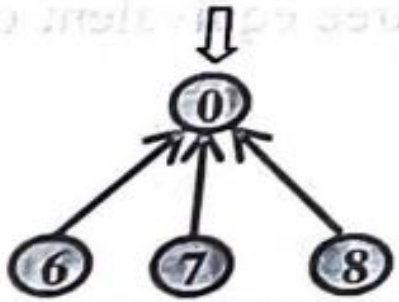
Find

Find(i) i is a node.

6.15.2 Find(i)

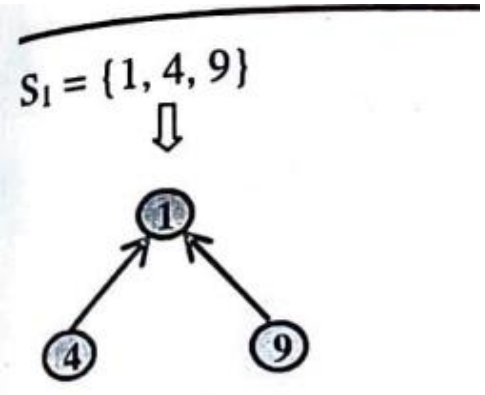
If i is a node in a tree, this function returns the root of the corresponding tree. For example, consider the sets :

$S_0 = \{0, 6, 7, 8\}$

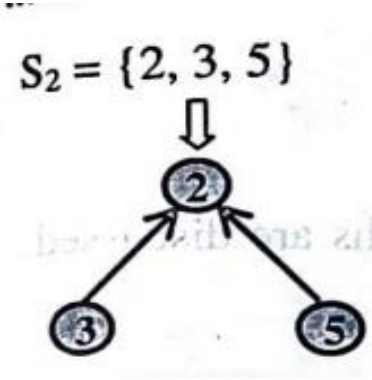


The root for node 6, node 7 and node 8 is 0. So, Find(6), Find(7) and Find(8) will be 0

Cont..



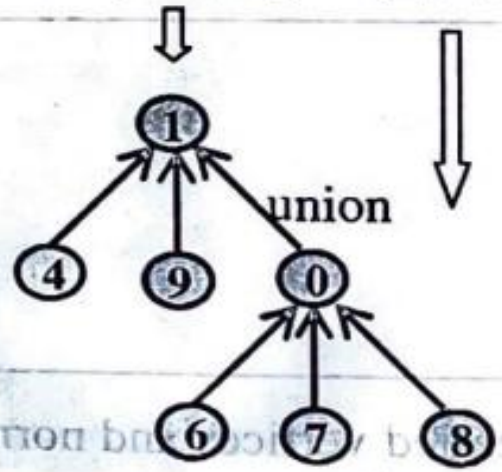
The root for node 4 and node 9 is 1. So, Find(4) and Find(9) will be 1



The root for node 3 and node 5 is 2. So, Find(3) and Find(5) will be 2

Array representation

$S_1 \cup S_0 = \{1, 4, 9\} \cup \{0, 6, 7, 8\}$



The root for node 4, node 9 and node 0 is node 1 whereas root for node 6, node 7 and node 8 is also node 1. The array representation of the tree is shown below:

s[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	-1			1		0	0	0	1

To find(8)

Note: Now, let us see, how to find(8). This can be done by starting from 8 as shown below: