

Construction of Binary Search Tree

Construction of Binary Search Tree and search

Sequential Algorithm

```
// Return a new node if the tree is empty
if (node == NULL) return newNode(key);

// Traverse to the right place and insert the node
if (key < node->key)
    node->left = insert(node->left, key);
else
    node->right = insert(node->right, key);

return node;
```

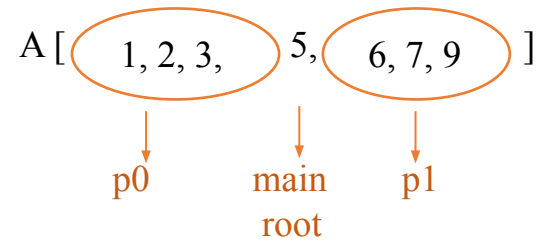
Parallel Algorithm

```
//Input is array 'a' contains n elements
// number of processors required=2
If len(a)>2
    root =a[i]
    parallel sort the given n elements [odd-even method]
    find the position of root in the sorted array
    if (root is at a[i] or a[n-1]) //either its a left tree or right tree
    {
        then split the array into half and provide it two processors
        call sequential insertion in each of processors
        broadcast the root node of both processors
    }
    else
    {
        provide the left and right of array elements of root to two processors
        call sequential insert in each processors
        broadcast the root node of both processors
    }
    combine the trees of both to the original root
```

Case 1

A[5, 2, 1, 3, 9, 7, 6]

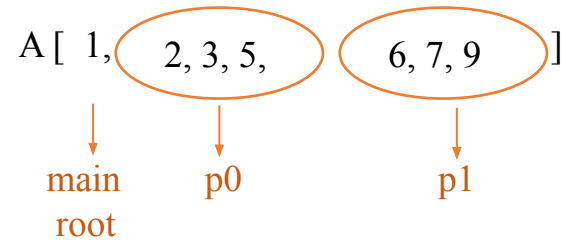
Sorted Array



Case 2

A[1, 2, 3, 5, 6, 7, 9]

Sorted Array



Case 3

A[9, 7, 6, 5, 3, 2, 1]

Sorted Array

