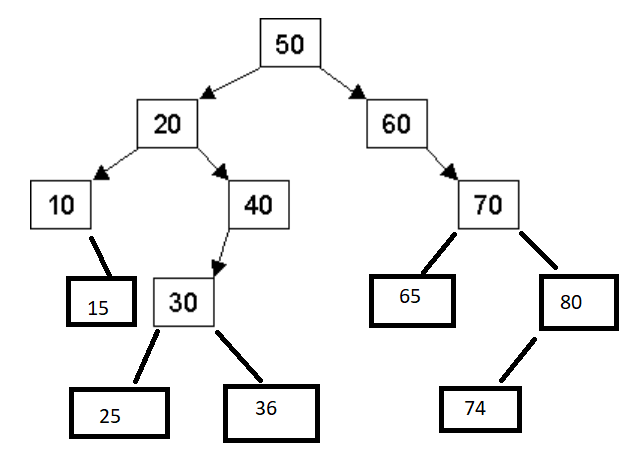
**Question 1a)**

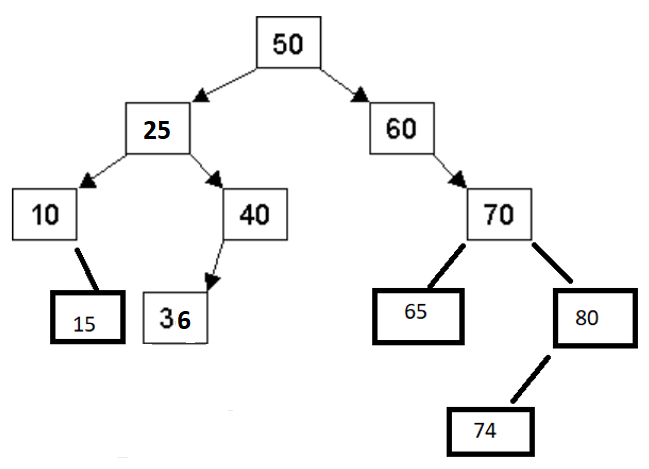


**Question 1b)**

Pre-Order:  
50-20-10-15-40-30-25-36-60-70-65-80-74  
  
In-Order:  
10-15-20-25-30-36-40-50-60-65-70-74-80

Post-Order:  
15-10-25-36-30-40-20-65-74-80-70-60-50

**Question 1c)**



**Question 2a)**

Struct Node {  
Int m\_value;  
Node \*m\_parent;  
Node \*m\_left, \*m\_right;  
};

**Question 2b)**

struct Node {

int m\_value;

node \*m\_parent;

node\* m\_left, \*m\_right;

//constructor

node(int value) {

m\_value = value;

m\_parent = m\_left = m\_right = nullptr;

}

};

void insert(const int &value, Node\* m\_root){

//If the root is empty, then create a node at this point

if (m\_root == nullptr){

m\_root = new Node(value);

}

//Create temp node pointing to the root

Node \*curr = m\_root;

//Keep repeating until you can insert the value

for(;;){

//If the value already exists at this node, return

if (value = curr->m\_value){return;}

//If the value you want to insert is less than the current Node's value

if (value < curr->m\_value) {

//If the left branch is not a nullptr

if (curr->m\_left != nullptr){

//Move curr to the next left branch

curr = curr->m\_left;

}

else {

//Create a new node called temp

Node \*temp = new Node(value);

//Set the left branch to this new node

curr->m\_left = temp;

//Update the parent pointer of this new node to the current node

temp->m\_parent = curr;

//Since we're done creating a new node, return

return;

}

}

//If the value you want to insert is greater than the current Node's value

if (value > curr->m\_value) {

//If the right branch is not a nullptr

if (curr->m\_right != nullptr){

//Move curr to the next right branch

curr = curr->m\_right;

}

else {

//Create a new node called temp

Node \*temp = new Node(value);

//Set the right branch to this new node

curr->m\_right = temp;

//Update the parent pointer of this new node to the current node

temp->m\_parent = curr;

//Since we're done creating a new node, return

return;

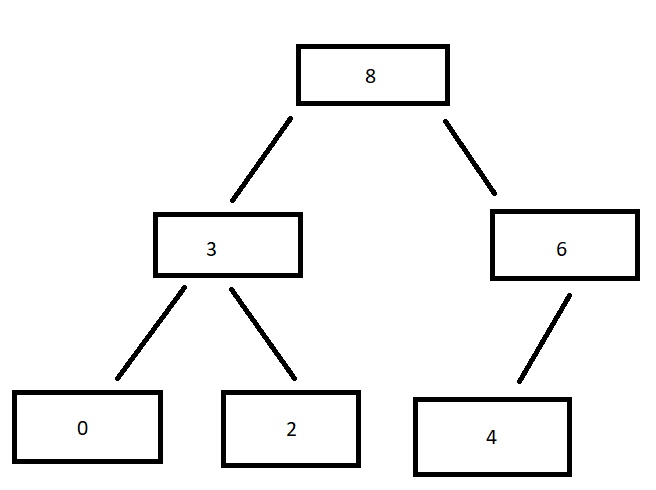
}

}

}

}

**Question 3a**



**Question 3b**

8,3,6,0,2,4

**Question 3c**

6,3,4,0,2

**Question 4**4a) O(C+S)  
4b) O(log C + S)  
4c) O (log C + log S) = O(log(CS))  
4d) Even distribution: O (log S)  
 Worst Case: O (C+log S)  
4e) Even Distribution O (1)  
 Worst Case: O (C+S)  
4f) O (log C + S)  
4g) Even Distribution: O (S log S)  
 Worst Case: O (C + S log S)  
4h) O (C log S)