Homework 5

Gautam Anand

405965589

1a.

50

20 60

10 40 70

15 30 68 80

26 35 75

1b.

Inorder

10,15,20,26,30,35,40,50,60,68,70,75,80

Preorder

50,20,10,15,40,30,26,35,60,70,68,80,75

Postorder

15,10,26,35,30,40,20,68,75,80,70,60,50

1c.

50

15 60

10 40 70

35 68 80

26 75

2a)

struct Node

{  
int number;

Node\* parent;

Node\* left;

Node\*right;

}

2b)

Void InsertNode(int value, Node\* current\_Node)

{  
If the current node is a nullptr, then create a new node and set the left and right and parent nodes to nullptr

Current points to this new node and return

Create a node pointer at the root of the tree

If the value that needs to be inserted is less than node value

If the left pointer exists, set current\_Node to the left node and call the function again

Else

Make a new node with the number inserted and set the parent node equal to the current\_Node

Set both the left and right pointers to nullptr

Set the Current Node left child to this new node

If the value that needs to be inserted is greater than node value

If the right pointer exists, set current\_Node to the right node and call the function again

Else

Make a new node with the number inserted and set the parent node equal to the current\_Node

Set both the left and right pointers to nullptr

Set the Current Node right child to this new node

Else

Both values are the same and do nothing, so return

}

3a)

7

5 6

1 0 3

3b)

[7,5,6,1,0,3]

3c)

Removes 7 in the heap

[6,5,3,1,0]

4)

1. Has to traverse through all the pairs, and there are C pairs, and then have to traverse through a sorted list which is O(S) complexity. Determining whether a student is enrolled in a course c, is **O(C+S)** complexity.
2. Has to traverse through a map, which is O(log(C)) complexity, and then traversing through a list is O(S) complexity. Determining whether a student is enrolled in a course c, is **O(log(C)+S)** complexity.
3. Have to traverse through a map, which is O(log(C)) complexity, and then traversing through a set is O(log(S)) complexity. Determining whether a student is enrolled in a course c, is **O(log(C)+log(S))** complexity.
4. Have to traverse through an unordered map, which is O(1) complexity, and then traversing through a set is O(log(S)) complexity. Determining whether a student is enrolled in a course c, is **O(1+log(S)) = O(log(S))**complexity.
5. Have to traverse through an unordered map, which is O(1) complexity, and then traversing through an unordered set is O(1) complexity. Determining whether a student is enrolled in a course c, is **O(1+1) = O(1)** complexity.
6. Have to traverse through a map, which is O(log(C)) complexity, and then traversing through a set to print out each element in the set is O(S) complexity. Determining whether a student is enrolled in a course c, is **O(log(C)+S)** complexity.
7. Have to traverse through an unordered map which is time complexity O(1) and another unordered set which is time complexity O(S) and then insert into a heap is O(log(S) and doing this s times is O(Slog(S)). Overall complexity is **O(1+S+S\*log(S)) = O(S\*logS)**
8. Have to traverse through all items in the unordered map which is time complexity O(C) and then traversing through a binary tree is O(log(S)). So the overall complexity is **O(C\*log(S))**