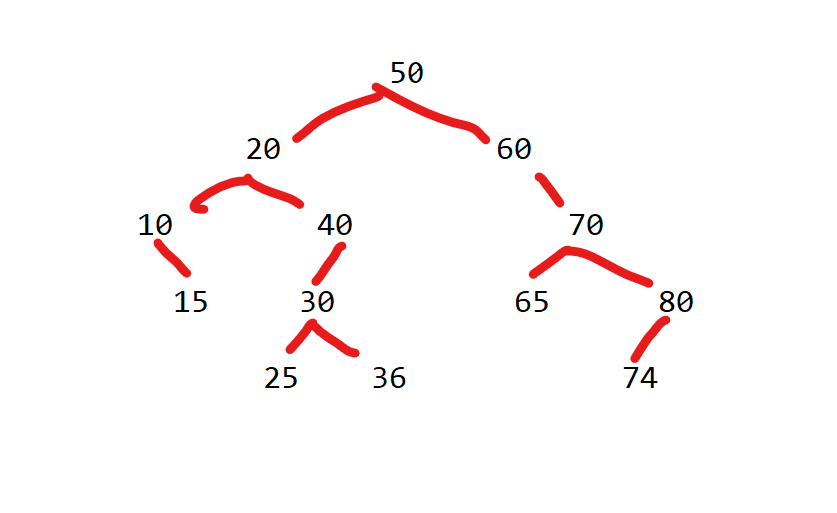
1a.



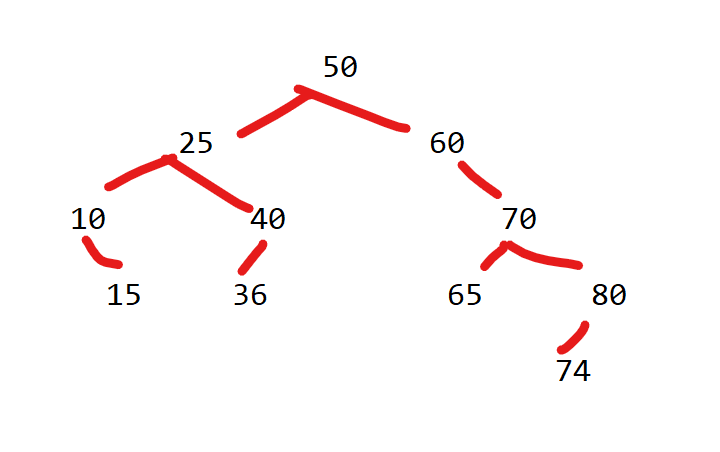
1b.

Inorder: 10, 15, 20, 25, 30, 36, 40, 50, 60, 5, 70, 74, 80

Preorder: 50, 20, 10, 15, 40, 30, 25, 36, 60, 70, 65, 80, 74

Postorder: 15, 10, 25, 36, 30, 40, 20, 65, 74, 80, 70, 60, 50

1c.



2a

struct Node

{

int value;

Node\* leftChild;

Node\* rightChild;

Node\* parent;

};

2b.

Psuedocode for insertion:

Given a certain value X,

Find the node with a parent pointer to null (that is the root)

Until insertion is complete

If X's value is equal to current node,

Do nothing, insertion complete

If X's value is less than current node

If there is a leftChild

Look at leftChild

Else

Create a new node with X's value

Set X's parent node to current node

Have current node's leftChild point to X

Set X's children nodes to null

Insertion complete

If X's value is greater than current node

If there is a rightChild

Look at rightChild

Else

Create a new node with X's value

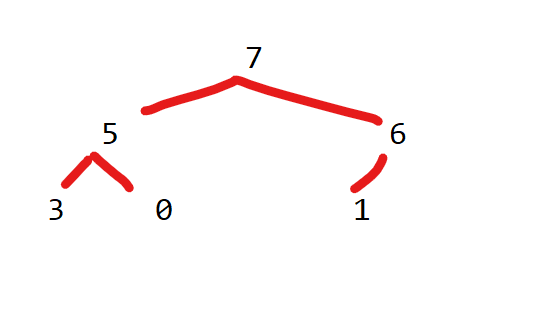
Set X's parent node to current

Have current node's rightChild pointer set to X

Set X's children to null

Insertion complete

3a.



3b

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | [0] | [1] | [2] | [3] | [4] | [5] |
| value | 7 | 5 | 6 | 3 | 0 | 1 |

3c.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | [0] | [1] | [2] | [3] | [4] |
| value | 6 | 5 | 1 | 3 | 0 |

4a. O(C + S)

4b. O(logC + S)

4c. O(logC + logS)

4d. O(logS)

4e. O(1)

4f. O(log c + s)

4g. O(c logS)

4h. O(s logS)