1a. Chart, line chart

Description automatically generated

1b. inorder

10

15

20

23

30

35

40

50

60

62

70

75

80

postorder:

15

10

23

35

30

40

20

62

75

80

70

60

50

preorder:

50

20

10

15

40

30

23

35

60

70

62

80

75

1c. Chart

Description automatically generated

struct Node{

Node(Node\* parent , int val)

:parent(parent), lChild(nullptr), rChild(nullptr), value(val)

{}

Node\* parent, \* lChild, \* rChild;

int value;

};

public:

void insert(int val){

else call recursiveInsert with parent as nullptr, current as m\_ root and val as value

}

private:

void recrusiveInsert(Node\* parent, Node\*& current, int val){

if current is nullptr, allocate new node at current with parent and val as parameters

otherwise,

if value equals current value, return immediately

if value is less than parent value, call recursive insert again with paremeters of current, pointer to left subtree, and value

if value is greater, instead pass a pointer to the right subtree

}

3a. A picture containing chart

Description automatically generated

3b.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 5 | 6 | 4 | 0 | 3 |  |  |  |  |

3c.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 5 | 3 | 4 | 0 |  |  |  |  |  |

Q4: using 1 as constant

4a. O(C + S)

4b. O(logC + S)

4c. O(logC + logS)

4d. O(1 + logS)

4e. O(1 + 1) -> O (1)

4f. O(logC + SlogS)

must find the correct object (logS) S amount of times

4g. O(1 + Slog(S))

4h. O(C + logS)