1)

20

50

65

75

80

70

60

35

25

40

10

15

30

50

20

10

40

15

30

25

35

60

70

65

80

75

b).

in-order:- 10, 15, 20, 25, 30, 35, 40, 50, 60, 65, 70, 75, 80,

pre-order: - 50, 20, 10, 15, 40, 30, 25, 35, 60, 70, 65, 80, 75

post-order:- 15, 10, 25, 35, 30, 40, 20, 65, 75, 80, 70, 60, 50

c). After deleting 30

20

50

65

75

80

70

60

25

40

10

15

35

Deleting node 20

1)

35

50

65

75

80

70

60

25

40

10

15

30

2. a)

 struct Node {

        //constructor for the node

        Node (cons int val) {

            value = val;

            left = right = NULL;

parent = NULL;

        }

        int value;

        Node \*left, \*right;

Node\* parent;

        //counts the total number of nodes

    };

b)

void insert(const int &value)

{

if m\_root is NULL

create a new node and then return

otherwise

create a current pointer node and assigned to root

used infinitely for loop

if the value is equal to current node value then return

if not,

if value is less than value of current node

if current left node is not NULL then increment the current pointer

otherwise

create a new node at the left node

then return

or if the value is greater than current node value

if the current right node is not NULL then increment the current node

otherwise

create a right node, then return

}

3.

a)

8

3

6

4

2

0

b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 8 | 3 | 6 | 0 | 2 | 4 |

c)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 3 | 4 | 0 | 2 |

4.

a. O(C+S)

b. O((logC + S)

c. O((logC) + log(S)

d. O(logS)

e.O(1)

f. O((logC) + S)

g.O(S\* log(S))

h.O(C\*(logS))