1.

a)

50

20 60

10 40 70

15 30 64 80

23 36 75

b)

inorder : 10 15 20 23 30 36 40 50 60 64 70 75 80

preorder : 50 20 10 15 40 30 23 36 60 70 64 80 75

postorder: 15 10 25 36 30 40 20 64 75 80 70 60 50

c)

After removing node 30:

50

20 60

10 40 70

15 36 64 80

23 75

After removing node 20:

50

23 60

10 40 70

15 36 64 80

75

2.

a)

struct Node

{

int value;

Node\* parent;

Node\* left;

Node\* right;

};

b)

If New's data is smaller than pos' data

Check if pos' left is null pointer

If true

set pos' left pointer to New and New's parent to pos

If false

Call insert recursively on the left child

If New's data is bigger than pos' data

Check if pos' right is null pointer

If true,

set pos' right poiter to New and New's parent to pos

If false

Call insert recursively on the right child

3.

a)

5

1 6

3 7

2

b)

(5, 1, 6, 3, 7, 2)

c)

(5, 1, 6, 3, 2)

4.

a) O(c+log(s))

b) O(log(c)+s)

c) O(log(c)+log(s))

d) O(log(s))

e) O(1)

f) O(log(c)+s)

g) O(s\*log(s))

h) O(c\*log(s))