**Homework 5 Solution**

**Q.1**

**a.**

50

20 60

10 40 70

15 30 65 80

25 39 71

**b.**

In-order – 10 15 20 25 30 39 40 50 60 65 70 71 80

Pre-order – 50 20 10 15 40 30 25 39 60 70 65 80 71

Post-order – 15 10 25 39 40 20 65 71 80 70 60 50

**c.**

After removing the 30

50

20 60

10 40 70

15 39 65 80

25 71

After removing the 20

50

25 60

10 40 70

15 39 65 80

71

**Q.2**

**a.**

struct Node

{

int value;

Node\* parent;

Node\* left;

Node\* right;

};

**b.**

void insert (Node\* pos, Node\* New)

{

If the New’s data is smaller than pos’ data

Check if the pos’ left child is a null pointer

If it is, pos’ left pointer to the New and the New’s parent to pos

If not

Call insert recursively on the left child

If the New’s data is larger than pos’ data

Check if the pos’ right child is a null pointer

If it is, set pos’ right pointer to the New and the New’s parent to pos

If not,

Call insert recursively on the right child

}

Note: I expect that the New is a well formed node which implies that its left and right pointers are nullptrs and the data value is valid.

**Q.3**

**a.**

7

3 6

0 2 4

**b.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 7 | 3 | 6 | 0 | 2 | 4 |

**c.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 6 | 3 | 4 | 0 | 2 |

**Q.4**

**a.** O(C+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)