Vanessa Lee

504766034

Homework 5

1. a.

50

20

60

70

40

10

80

65

30

15

25

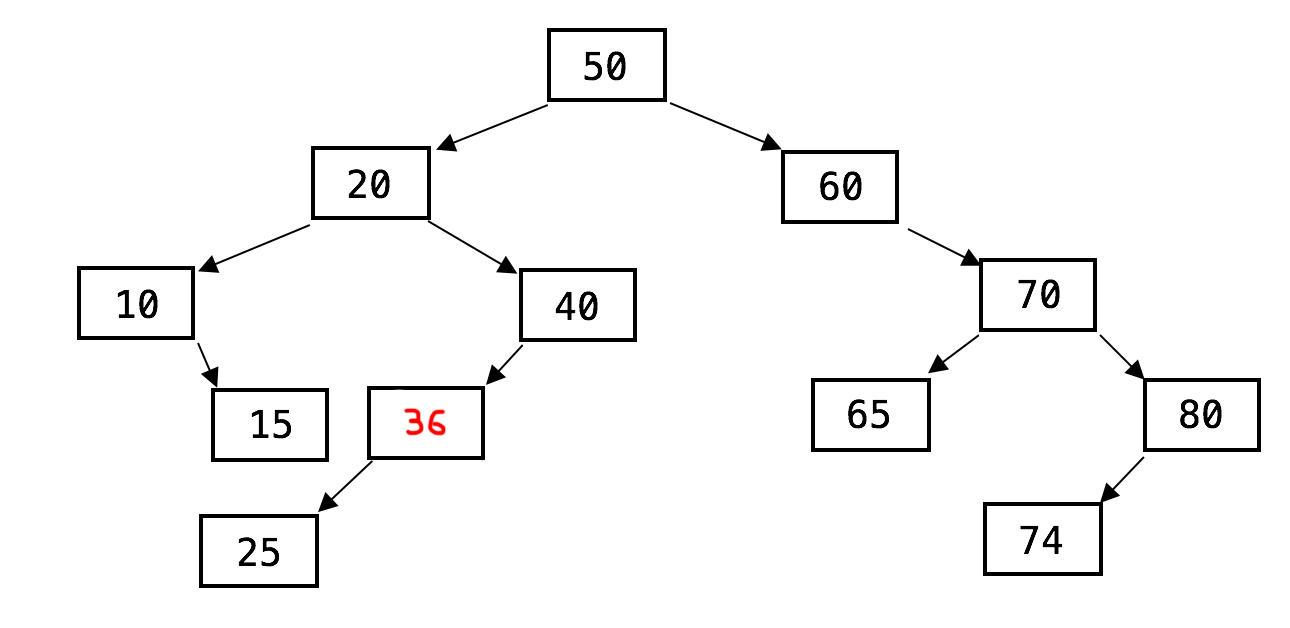
74

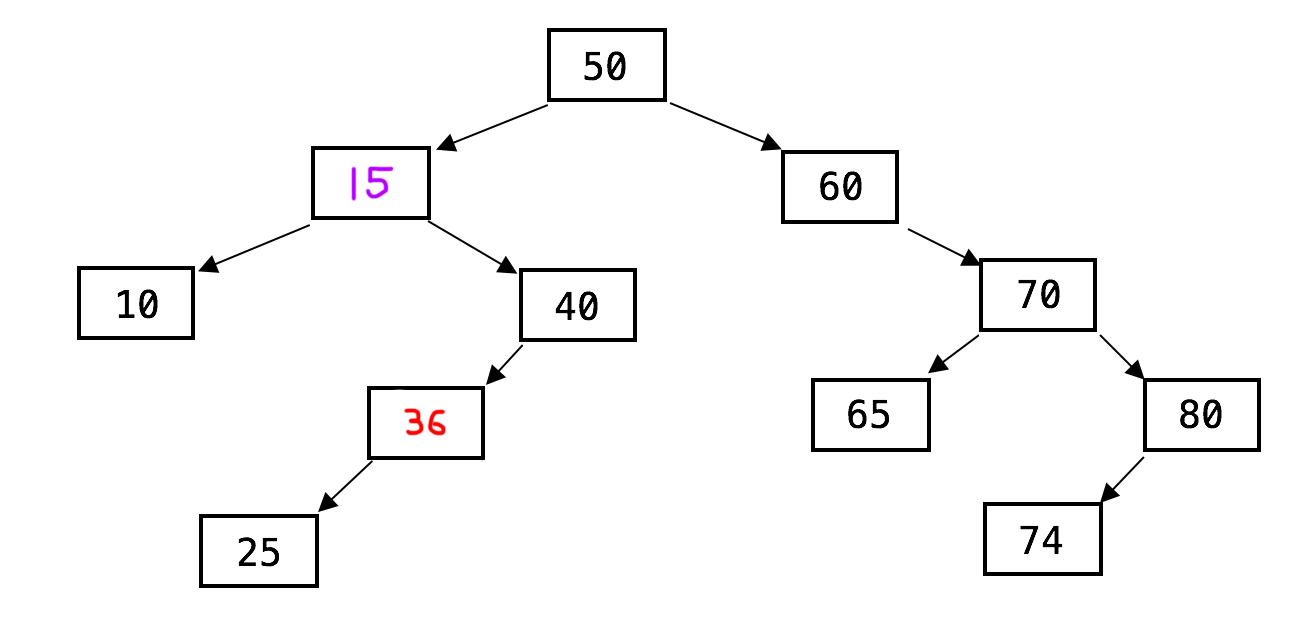
36

b. In-Order: 10 15 20 25 30 36 50 60 65 70 74 80

Pre-Order: 50 20 10 15 40 30 25 36 60 70 65 80 74

Post-Order: 15 10 25 36 30 40 65 74 80 70 60 50

c.



1. a.

struct BinaryTree

{

BinaryTree(int i)

{

data = i;

left = nullptr;

right = nullptr;

parent = nullptr;

}

int data;

BinaryTree\* left;

BinaryTree\* right

BinaryTree\* parent;

}

b.

insert(BinaryTree\* node, BinaryTree\* root)

If root is nullptr, set root to point to new node and set parent pointer to nullptr

If node's value is smaller than root's value, go left

If there is no left child, set left pointer to new node and set parent pointer to root

Else recursively call insert function with left child as root

else if node's value is greater than root's value, go right

If there is no right child, set right pointer to new node and set parent pointer to root

Else recursively call insert function with right child as root

1. a.

8

6

3

0

2

4

|  |  |
| --- | --- |
| 0 | 8 |
| 1 | 3 |
| 2 | 6 |
| 3 | 0 |
| 4 | 2 |
| 5 | 4 |

b.

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

c.

1. 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(1 + S)

h. O(C \* log S)