Calvin Liu

CS 32 Hwk 5

50

1a)

20

60

70

10

40

45

65

80

30

25

35

75

1b)

50

60

70

25

40

10

35

45

65

80

75

1c)

In-order would print = 10, 20, 25, 30, 35, 40, 45, 50, 60, 65, 70, 75, 80

Pre-order would print = 50, 20, 10, 40, 30, 25, 35, 45, 60, 70, 65, 80, 75

Post-order would print = 10, 25, 35, 30, 45, 40, 20, 65, 75, 80, 70, 60, 50

2a)

8

3

5

4

2

1

b) [8, 3, 5, 1, 2, 4]

c) [5, 3, 4, 1, 2]

3a)

struct Node\*

{

int m\_value;

Node\* m\_parent;

Node\* m\_right;

Node\* m\_left;

}

3b)

current should be the root

parent should be NULL

insert(current, parent, value)

if it is empty, there’s no root

make the root equal to the new node

set its value

make its left and right children NULL

make the parent pointer NULL

if the value is less than the current’s value

if the current’s left is NULL

current’s left equal to insert a new node

make current’s left left NULL

make current’s left right NULL

new node’s value will equal value

new node’s parent will be root

else if the value is greater than or equal to the root’s value

if the current’s right is NULL

current’s right equal to insert a new node

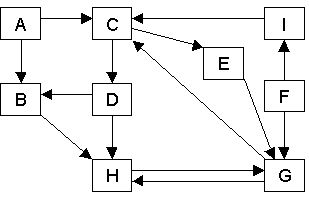
make current’s left left NULL

make current’s left right NULL

new node’s value will equal value

new node’s parent will be root

insert(root’s right, root, value)

4a)

From To

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | | G | H | I |
| A | 0 | 1 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 1 | 0 |
| C | 0 | 0 | 0 | 1 | 1 | 0 | | 0 | 0 | 0 |
| D | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 1 | 0 |
| E | 0 | 0 | 0 | 0 | 0 | 0 | | 1 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 0 | | 1 | 0 | 1 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 1 | 1 |
| H | 0 | 0 | 0 | 0 | 0 | 0 | | 1 | 0 | 0 |
| I | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 0 |

List:

A -> B and C

B -> H

C -> E and D

D -> B and H

E -> G

F -> G and I

G -> H and C

H -> G

I -> C

4b)

Starting from E:

EGHCDB

EGCDBH

EGCDHB