Justin S Lee

CS 32 – 804603677

June 4, 2018

Homework 5

1. a.

73

80

65

37

25

30

15

70

10

40

20

60

50

b. in-order: 10 15 20 25 30 37 40 50 65 70 73 80

pre-order: 50 20 10 15 40 30 25 37 60 70 65 80 73

post-order: 15 10 25 37 30 40 20 65 80 73 70 60 50

c.

73

80

65

37

15

70

10

40

25

60

50

1. a. structure for a binary tree node

struct Node

{

node (const string &i\_data)

{

m\_data = i\_data;

left = right = parent = nullptr;

}

int m\_data;

node \*left, \*right, \*parent

}

b. Pseudocode to insert a new node into a binary search tree with parent pointers

if the tree is empty

have the root of the binary search tree be equal to a construction of a new node with some value and return

initialize two pointers, cur and parent, which are equal to the root node

in an infinite loop

if the value is equal to cur’s value, then return

if the value is less than cur’s value and cur’s left is not a nullptr, then have parent equal to cur and cur move to cur’s left

else cur’s left is equal to the construction of a new node and its parent is equal to the initialized parent node

if the value is greater than cur’s value and cur’s right is not a nullptr, then have parent equal to cur and cur move to cur’s right

else cur’s right is equal to the construction of a new node and its parent is equal to the initialized parent node

1. a.

0

2

3

6

8

4

b.

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

c.

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

1. a. O (C+S)

b. O (log2C + S)

c. O (log2C + log2S)

d. O (log2S)

e. O (1)

f. O (log2C + S)

g. O (S2 log2 S)

h. O (C log2S)