HW #5

1.

a. Resulting tree

50

20

60

10

40

70

80

15

65

30

76

34

25

b.

in-order traversal: 10 15 20 25 30 34 40 50 60 65 70 76 80

pre-order traversal: 50 20 10 15 40 30 25 34 60 70 65 80 76

post-order traversal: 15 10 25 34 30 40 20 65 76 80 70 60 50

c.

After deleting node 30:

50

20

60

10

40

70

80

15

65

34

76

25

And then deleting node 20:

50

25

60

10

40

70

80

15

65

34

76

2.

a.

struct Node {

int data;

Node\* parent;

Node\* left;

Node\* right;

}

b. Pseudocode for BST insertion

If the parent is a nullptr (tree is empty), then we set parent to be the node to insert.

Otherwise, we do the following:

while we haven’t inserted yet:

If the node to insert’s data is less than the parent

If there is a left child, go to that child and set it as the current parent

else: insert the node to insert as the parent’s left child

If the node to insert’s data is greater than the parent

If there is a right child, go to that child and set it as the current parent

else: insert the node to insert as the parent’s right child

3.

a. Resulting heap:

7

3

5

0

4

2

b.

Array form:

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

c. h.remove(item) and resulting array:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 3 | 4 | 0 | 2 |

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(Slog(S))

We take O(1) to find the course. Then, we sort the unordered set with mergesort. This is O(Slog(S)). Then, printing the items is O(S). But, O(Slog(S) + S) simplifies to O(Slog(S)).

h. O(Clog(S))

Must go through every course O(C), and for every course it needs to find the student O(log(S)).

5.

a. list.cpp

b. Given the constraints in part a, I could not solve this problem with only a one-parameter listAll because we wouldn’t be able to print the statements properly. They would misaligned since recursion would take the print statements to the lowest level and then back up, and then it wouldn’t go one level at a time. It would go through an entire subclass list before moving onto the next, in all likelihood. With the two-parameter listAll, we are able to keep track of the string that we want printed and how many times we print each thing.