Hanna Co

1.

50



60

25

30

34

40

15

10

80

70

20



65

76

b.

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

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

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

c.



50



60

25

34

40

10

80

70

15



65



76

2. struct Node

{

private:

int value;

Node\* parent;

Node\* left;

Node\* right;

};

b. if left and right are both nullptr

if parent is nullptr

delete the target node

else

set the parent’s appropriate node (right or left) to null and delete the target node

else if either left or right are nullptr

if parent is nullptr

set the child’s parent to nullptr

delete the target node

else

set the target’s child node’s parent to the target node’s parent

delete the target node

else

choose the left subtree’s largest child or the right subtree’s smallest child

link that node to the target’s parent and the target’s child, and unlink it from its original parent and re-link any children it has

delete the target node

3.



40

5

7

3

2

0

b.

4

2

5

3

0

7

c.

2

0

3

4

5

4. a. O(C + log(S))

b. O(logC + S)

c. O(logC + logS)

d. O(C + logS)

e. O(C + S)

f. O(log C + S)

g. O(C + S^2)

h. O(C\*log(S))

5. b. There would be no way to access the name of the superclass, because with every recursive call, any string you made would be re-created.