

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 class BST {
7 public:
8     int value;
9     BST *left;
10    BST *right;
11
12    BST(int val) {
13        value = val;
14        left = NULL;
15        right = NULL;
16    }
17
18    // Average: O(log(n)) time | O(1) space
19    // Worst: O(n) time | O(1) space
20    BST &insert(int val) {
21        BST *currentNode = this;
22        while (true) {
23            if (val < currentNode->value) {
24                if (currentNode->left == NULL) {
25                    BST *newNode = new BST(val);
26                    currentNode->left = newNode;
27                    break;
28                } else {
29                    currentNode = currentNode->left;
30                }
31            } else {
32                if (currentNode->right == NULL) {
33                    BST *newNode = new BST(val);
34                    currentNode->right = newNode;
35                    break;
36                } else {
37                    currentNode = currentNode->right;
38                }
39            }
40        }
41        return *this;
42    }
43
44    // Average: O(log(n)) time | O(1) space
45    // Worst: O(n) time | O(1) space
46    bool contains(int val) {
47        BST *currentNode = this;
48        while (currentNode != NULL) {
49            if (val < currentNode->value) {
50                currentNode = currentNode->left;
51            } else if (val > currentNode->value) {
52                currentNode = currentNode->right;
53            } else {
54                return true;
55            }
56        }
57        return false;
58    }
59
60    // Average: O(log(n)) time | O(1) space
61    // Worst: O(n) time | O(1) space
62    BST &remove(int val, BST *parentNode = NULL) {
63        BST *currentNode = this;
64        while (currentNode != NULL) {
65            if (val < currentNode->value) {
66                parentNode = currentNode;
67                currentNode = currentNode->left;
68            } else if (val > currentNode->value) {
69                parentNode = currentNode;
70                currentNode = currentNode->right;
71            } else {
72                if (currentNode->left != NULL && currentNode->right != NULL) {
73                    currentNode->value = currentNode->right->getMinValue();
74                    currentNode->right->remove(currentNode->value, currentNode);
75                } else if (parentNode == NULL) {
76                    if (currentNode->left != NULL) {
77                        currentNode->value = currentNode->left->value;
78                        currentNode->right = currentNode->left->right;
79                        currentNode->left = currentNode->left->left;
80                    } else if (currentNode->right != NULL) {
81                        currentNode->value = currentNode->right->value;
82                        currentNode->left = currentNode->right->left;
83                        currentNode->right = currentNode->right->right;
84                    } else {
85                        // This is a single-node tree; do nothing.
86                    }
87                } else if (parentNode->left == currentNode) {
88                    parentNode->left = currentNode->left != NULL ? currentNode->left
89                        : currentNode->right;
90                } else if (parentNode->right == currentNode) {
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 // Do not edit the class below except for
5 // the insert, contains, and remove methods.
6 // Feel free to add new properties and methods
7 // to the class.
8 class BST {
9 public:
10    int value;
11    BST *left;
12    BST *right;
13
14    BST(int val) {
15        value = val;
16        left = NULL;
17        right = NULL;
18    }
19
20    BST &insert(int val) {
21        // Write your code here.
22        // Do not edit the return statement of this method.
23        return *this;
24    }
25
26    bool contains(int val) {
27        // Write your code here.
28        return false;
29    }
30
31    BST &remove(int val) {
32        // Write your code here.
33        // Do not edit the return statement of this method.
34        return *this;
35    }
36 };
37
```

Custom Output

Raw Output

Submit Code

```
91         parentNode->right = currentNode->left != NULL ? currentNode->left
92         : currentNode->right;
93     }
94     break;
95 }
96 }
97 return *this;
98 }
99
100 int getMinValue() {
101     if (left == NULL) {
102         return value;
103     } else {
104         return left->getMinValue();
105     }
106 }
107 };
108
```

Run or submit code when you're ready.