

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <deque>
5 using namespace std;
6
7 class Node {
8 public:
9     string name;
10    vector<Node*> children;
11
12    Node(string name) { this->name = name; }
13
14    // O(v + e) time | O(v) space
15    vector<string> breadthFirstSearch(vector<string> *array) {
16        deque<Node*> queue{this};
17        while (!queue.empty()) {
18            Node current = *queue.front();
19            queue.pop_front();
20            array->push_back(current.name);
21            for (int i = 0; i < current.children.size(); i++) {
22                queue.push_back(current.children[i]);
23            }
24        }
25        return *array;
26    }
27
28    Node *addChild(string name) {
29        Node *child = new Node(name);
30        children.push_back(child);
31        return this;
32    }
33 };
34
```

Solution 1 Solution 2 Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 // Do not edit the class below except
5 // for the breadthFirstSearch method.
6 // Feel free to add new properties
7 // and methods to the class.
8 class Node {
9 public:
10    string name;
11    vector<Node*> children;
12
13    Node(string str) { name = str; }
14
15    vector<string> breadthFirstSearch(vector<string> *array) {
16        // Write your code here.
17        return {};
18    }
19
20    Node *addChild(string name) {
21        Node *child = new Node(name);
22        children.push_back(child);
23        return this;
24    }
25 };
26
```

Run or submit code when you're ready.