AlgoExpert

**Quad Layout** 

C:

12px

Sublime

Monokai

00:00:

Our Solution(s) Run

```
Run Code
```

Your Solutions

Run Code

```
Solution 1
```

```
_{\rm 1} // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   using System.Collections.Generic;
    public class Program {
     public class Node {
        public string name;
        public List<Node> children = new List<Node>();
        public Node(string name) {
         this.name = name;
13
14
        // O(v + e) time | O(v) space
        public List<string> BreadthFirstSearch(List<string> array) {
16
         Queue<Node> queue = new Queue<Node>();
         queue.Enqueue(this);
17
18
         while (queue.Count > 0) {
           Node current = queue.Dequeue();
19
            array.Add(current.name);
20
           current.children.ForEach(o => queue.Enqueue(o));
         return array;
24
25
26
        public Node AddChild(string name) {
27
         Node child = new Node(name);
         children.Add(child);
28
         return this;
```

```
1 using System.Collections.Generic;
```

Solution 1 Solution 2

```
public class Program {
      // Do not edit the class below except
      \label{lem:continuous} \mbox{// for the BreadthFirstSearch method.}
      // Feel free to add new properties
      // and methods to the class.
      public class Node {
        public string name;
10
        public List<Node> children = new List<Node>();
        public Node(string name) {
13
         this.name = name;
14
16
        public List<string> BreadthFirstSearch(List<string> array) {
         // Write your code here.
18
          return null;
19
20
        public Node AddChild(string name) {
          Node child = new Node(name);
22
23
          children.Add(child);
24
          return this;
25
26
27 }
28
```

Solution 3

Custom Output Raw Output Submit Code

Run or submit code when you're ready.