

Our Solution(s)

Run Code

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 class TrieNode {
8 public:
9     unordered_map<char, TrieNode *> children;
10 };
11
12 class ModifiedSuffixTrie {
13 public:
14     TrieNode *root;
15
16     ModifiedSuffixTrie(string str) {
17         this->root = new TrieNode();
18         this->populateModifiedSuffixTrieFrom(str);
19     }
20
21     void populateModifiedSuffixTrieFrom(string str) {
22         for (int i = 0; i < str.length(); i++) {
23             this->insertSubstringStartingAt(i, str);
24         }
25     }
26
27     void insertSubstringStartingAt(int i, string str) {
28         TrieNode *node = this->root;
29         for (int j = i; j < str.length(); j++) {
30             char letter = str[j];
31             if (node->children.find(letter) == node->children.end()) {
32                 TrieNode *newNode = new TrieNode();
33                 node->children.insert({letter, newNode});
```

Your Solutions

Run Code

Solution 1Solution 2Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<bool> multiStringSearch(string bigString, vector<string> smallStrings) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <string>
5
6 using namespace std;
7
8 // Returns true if bigString contains all of smallStrings as a substring.
9 bool multiStringSearch(string bigString, vector<string> smallStrings) {
10     // Write your code here.
11 }
```

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <string>
5
6 using namespace std;
7
8 // Returns true if bigString contains all of smallStrings as a substring.
9 bool multiStringSearch(string bigString, vector<string> smallStrings) {
10     // Write your code here.
11 }
```

```
10         "What's the best way to get to the airport?"
11     )
12
13     # Print the output of the function
14     print(best_trip_time(
15         start="New York",
16         end="Los Angeles",
17         mode="air",
18         departure_time="2023-01-01T08:00:00",
19         arrival_time="2023-01-01T18:00:00",
20         departure_date="2023-01-01",
21         arrival_date="2023-01-01",
22         departure_time_zone="America/New_York",
23         arrival_time_zone="America/Los_Angeles",
24     ))
25
26     # Print the output of the function
27     print(best_trip_time(
28         start="New York",
29         end="Los Angeles",
30         mode="air",
31         departure_time="2023-01-01T08:00:00",
32         arrival_time="2023-01-01T18:00:00",
33         departure_date="2023-01-01",
34         arrival_date="2023-01-01",
35         departure_time_zone="America/New_York",
36         arrival_time_zone="America/Los_Angeles",
37     ))
```

Run or submit code when you're ready.