AlgoExpert Quad Layout C++ 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2 Solution 3

60

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   #include <vector>
   #include <unordered_map>
   using namespace std;
   class TrieNode {
   public:
     unordered_map<char, TrieNode *> children;
10 };
11
12
    class ModifiedSuffixTrie {
13
   public:
14
     TrieNode *root;
15
      ModifiedSuffixTrie(string str) {
16
17
       this->root = new TrieNode();
       this->populateModifiedSuffixTrieFrom(str);
18
19
20
      void populateModifiedSuffixTrieFrom(string str) {
21
22
       for (int i = 0; i < str.length(); i++) {</pre>
23
         this->insertSubstringStartingAt(i, str);
24
25
26
27
      void insertSubstringStartingAt(int i, string str) {
28
        TrieNode *node = this->root;
29
        30
         char letter = str[j];
         if (node->children.find(letter) == node->children.end()) {
31
32
           TrieNode *newNode = new TrieNode();
           node->children.insert({letter, newNode});
33
34
         node = node->children[letter];
35
36
37
38
39
      bool contains(string str) {
40
       TrieNode *node = this->root;
        for (char letter : str) {
41
         if (node->children.find(letter) == node->children.end()) {
42
43
           return false;
44
45
         node = node->children[letter];
46
47
48
49
   };
50
51
   // O(b^2 + ns) time | O(b^2 + n) space
    vector<bool> multiStringSearch(string bigString, vector<string> smallStrings) {
52
      ModifiedSuffixTrie modifiedSuffixTrie(bigString);
53
54
     vector<bool> solution;
55
      for (string smallString : smallStrings) {
56
       solution.push_back(modifiedSuffixTrie.contains(smallString));
57
     return solution;
59
   }
```