

Our Solution(s)

Run Code

Solution 1

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

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 #include <unordered_map>
2 using namespace std;
3
4 // Do not edit the class below except for the
5 // populateSuffixTrieFrom and contains methods.
6 // Feel free to add new properties and methods
7 // to the class.
8 class TrieNode {
9 public:
10     unordered_map<char, TrieNode *> children;
11 };
12
13 class SuffixTrie {
14 public:
15     TrieNode *root;
16     char endSymbol;
17
18     SuffixTrie(string str) {
19         this->root = new TrieNode();
20         this->endSymbol = '*';
21         this->populateSuffixTrieFrom(str);
22     }
23
24     void populateSuffixTrieFrom(string str) {
25         // Write your code here.
26     }
27
28     bool contains(string str) {
29         // Write your code here.
30         return false;
31     }
32 };
33
```

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string s;
6      cin >> s;
7      int n = s.length();
8      int m = 0;
9      int k = 0;
10     for (int i = 0; i < n; i++) {
11         if (s[i] == '0') {
12             m++;
13             if (m == k) {
14                 k++;
15                 m = 0;
16             }
17         }
18     }
19     cout << k << endl;
20     return 0;
21 }

```

Run or submit code when you're ready.