Sublime AlgoExpert **Quad Layout 12px**

Video Explanation

00:00:00 Monokai

Solution 1

Scratchpad

Our Solution(s)

Prompt

75 } 76

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 #include <vector>
 4 #include <numeric>
 5 using namespace std;
 7 vector<vector<int>> getLocations(string str, string subStr);
 8 vector<vector<int>> collapse(vector<vector<int>> locations);
9 string underscorify(string str, vector<vector<int>>> locations);
11 // O(n*m) | O(n) space
12 string underscorifySubstring(string str, string subStr) {
13
     vector<vector<int>> locations = collapse(getLocations(str, subStr));
14
     return underscorify(str, locations);
15
16
17 vector<vector<int>>> getLocations(string str, string subStr) {
18
      vector<vector<int>> locations{};
     int startIdx = 0;
19
20
      while (startIdx < str.length()) {</pre>
21
        int nextIdx = str.find(subStr, startIdx);
22
        if (nextIdx != string::npos) {
23
          locations.push_back(vector<int>{nextIdx, int(nextIdx + subStr.length())});
24
          startIdx = nextIdx + 1;
25
        } else {
26
          break;
27
28
29
      return locations;
30 }
31
32
   vector<vector<int>> collapse(vector<vector<int>> locations) {
33
      if (locations.empty()) {
34
       return locations;
35
36
      \verb|vector<|int>> | newLocations{|locations[0]|};
37
      vector<int> *previous = &newLocations[0];
      for (int i = 1; i < locations.size(); i++) {</pre>
38
        vector<int> *current = &locations[i];
39
40
        if (current->at(0) <= previous->at(1)) {
41
          previous->at(1) = current->at(1);
42
43
          newLocations.push_back(*current);
44
          previous = &newLocations[newLocations.size() - 1];
45
46
47
      return newLocations;
48 }
49
   string underscorify(string str, vector<vector<int>>> locations) {
51
      int locationsIdx = 0;
52
      int stringIdx = 0;
53
      bool inBetweenUnderscores = false;
     vector<string> finalChars{};
54
55
      int i = 0;
56
      while (stringIdx < str.length() && locationsIdx < locations.size()) {</pre>
57
        if (stringIdx == locations[locationsIdx][i]) {
58
          finalChars.push_back("_");
59
          inBetweenUnderscores = !inBetweenUnderscores;
60
          if (!inBetweenUnderscores) {
61
            locationsIdx++;
62
63
          i = i == 1 ? 0 : 1;
64
        string s(1, str[stringIdx]);
65
66
        finalChars.push_back(s);
67
        stringIdx++;
68
69
      if (locationsIdx < locations.size()) {</pre>
70
       finalChars.push_back("_");
71
      } else if (stringIdx < str.length()) {</pre>
72
        finalChars.push_back(str.substr(stringIdx));
73
74
      return accumulate(finalChars.begin(), finalChars.end(), string());
```

Run Code