

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 bool isInBigString(string bigString, string smallString);
7 bool isInBigStringHelper(string bigString, string smallString, int startIdx);
8
9 // O(bns) time | O(n) space
10 vector<bool> multiStringSearch(string bigString, vector<string> smallStrings) {
11     vector<bool> solution;
12     for (string smallString : smallStrings) {
13         solution.push_back(isInBigString(bigString, smallString));
14     }
15     return solution;
16 }
17
18 bool isInBigString(string bigString, string smallString) {
19     for (int i = 0; i < bigString.length(); i++) {
20         if (i + smallString.length() > bigString.length()) {
21             break;
22         }
23         if (isInBigStringHelper(bigString, smallString, i)) {
24             return true;
25         }
26     }
27     return false;
28 }
29
30 bool isInBigStringHelper(string bigString, string smallString, int startIdx) {
31     int leftBigIdx = startIdx;
32     int rightBigIdx = startIdx + smallString.length() - 1;
33     int leftSmallIdx = 0;
34     int rightSmallIdx = smallString.length() - 1;
35     while (leftBigIdx <= rightBigIdx) {
36         if (bigString[leftBigIdx] != smallString[leftSmallIdx] ||
37             bigString[rightBigIdx] != smallString[rightSmallIdx]) {
38             return false;
39         }
40         leftBigIdx++;
41         rightBigIdx--;
42         leftSmallIdx++;
43         rightSmallIdx--;
44     }
45     return true;
46 }
47
```