

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 vector<int> buildPattern(string substr);
7 bool doesMatch(string str, string substr, vector<int> pattern);
8
9 // O(n + m) time | O(m) space
10 bool knuthMorrisPrattAlgorithm(string str, string substr) {
11     vector<int> pattern = buildPattern(substr);
12     return doesMatch(str, substr, pattern);
13 }
14
15 vector<int> buildPattern(string substr) {
16     vector<int> pattern(substr.size(), -1);
17     int j = 0;
18     int i = 1;
19     while (i < substr.size()) {
20         if (substr[i] == substr[j]) {
21             pattern[i] = j;
22             i++;
23             j++;
24         } else if (j > 0) {
25             j = pattern[j - 1] + 1;
26         } else {
27             i++;
28         }
29     }
30     return pattern;
31 }
32
33 bool doesMatch(string str, string substr, vector<int> pattern) {
34     int i = 0;
35     int j = 0;
36     while (i + substr.size() - j <= str.size()) {
37         if (str[i] == substr[j]) {
38             if (j == substr.size() - 1)
39                 return true;
40             i++;
41             j++;
42         } else if (j > 0) {
43             j = pattern[j - 1] + 1;
44         } else {
45             i++;
46         }
47     }
48     return false;
49 }
```

