

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n + m) time | O(m) space
5     func knuthMorrisPrattAlgorithm(_ string: String, subString: String) -> Bool {
6         let pattern = buildPattern(subString)
7         return doesMatch(string, subString, pattern)
8     }
9
10    func buildPattern(_ subString: String) -> [Int] {
11        var pattern = Array(repeating: -1, count: subString.count)
12
13        var i = 1
14        var j = 0
15
16        while i < subString.count {
17            let iStringIndex = subString.index(subString.startIndex, offsetBy: i)
18            let jStringIndex = subString.index(subString.startIndex, offsetBy: j)
19
20            if subString[iStringIndex] == subString[jStringIndex] {
21                pattern[i] = j
22
23                i += 1
24                j += 1
25            } else if j > 0 {
26                j = pattern[j - 1] + 1
27            } else {
28                i += 1
29            }
30        }
31
32        return pattern
33    }
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func knuthMorrisPrattAlgorithm(_ string: String, subString: String) -> Bool {
3         // Write your code here.
4         return false
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func knuthMorrisPrattAlgorithm(_ string: String, subString: String) -> Bool {
3         // Write your code here.
4         return false
5     }
6 }
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n + m) time | O(m) space
5     func knuthMorrisPrattAlgorithm(_ string: String, subString: String) -> Bool {
6         let pattern = buildPattern(subString)
7         return doesMatch(string, subString, pattern)
8     }
9
10    func buildPattern(_ subString: String) -> [Int] {
11        var pattern = Array(repeating: -1, count: subString.count)
12
13        var i = 1
14        var j = 0
15
16        while i < subString.count {
17            let iStringIndex = subString.index(subString.startIndex, offsetBy: i)
18            let jStringIndex = subString.index(subString.startIndex, offsetBy: j)
19
20            if subString[iStringIndex] == subString[jStringIndex] {
21                pattern[i] = j
22
23                i += 1
24                j += 1
25            } else if j > 0 {
26                j = pattern[j - 1] + 1
27            } else {
28                i += 1
29            }
30        }
31
32        return pattern
33    }
```

