**Your Solutions** 

Run Code

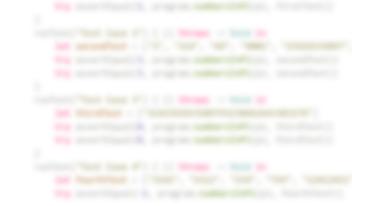
Our Solution(s) Run Code

```
Solution 1 Solution 2
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 3
    class Program {
        // O(n^3 + m) time | O(n + m) space
 4
        func numbersInPi(_ pi: String, _ favoriteNumbers: [String]) -> In
            var numbersDictionary = [String: Bool]()
            for number in favoriteNumbers {
 9
                numbersDictionary[number] = true
10
11
12
            var cache = [Int: Int]()
13
14
            for i in stride(from: pi.count - 1, through: 0, by: -1) {
                 getMinimumNumberOfSpaces(pi, numbersDictionary, &cache, i)
16
17
            \textbf{if} \ \mathsf{cache}[\texttt{0}] \ \texttt{==} \ \mathsf{Int}(\mathsf{Int32.max}) \ \{
18
19
                return -1
20
            } else {
21
                return cache[0]!
22
23
24
25
        func getMinimumNumberOfSpaces(_ pi: String, _ numbersDictionary: [
26
            if index == pi.count {
27
                return -1
28
29
30
            if let minimumNumberOfSpaces = cache[index] {
31
                 return minimumNumberOfSpaces
32
```

-

---

33



Run or submit code when you're ready.