

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^3 + m) time | O(n + m) space
5     func numbersInPi(_ pi: String, _ favoriteNumbers: [String]) -> Int {
6         var numbersDictionary = [String: Bool]()
7
8         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) {
15            getMinimumNumberOfSpaces(pi, numbersDictionary, &cache, i)
16        }
17
18        if cache[0] == Int(Int32.max) {
19            return -1
20        } else {
21            return cache[0]!
22        }
23    }
24
25    func getMinimumNumberOfSpaces(_ pi: String, _ numbersDictionary: [String: Bool], _ cache: [Int: Int]) {
26        if index == pi.count {
27            return -1
28        }
29
30        if let minimumNumberOfSpaces = cache[index] {
31            return minimumNumberOfSpaces
32        }
33    }
34}
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func numbersInPi(_ pi: String, _ favoriteNumbers: [String]) -> Int {
3         // Write your code here.
4         return -1
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^3 + m) time | O(n + m) space
5     func numbersInPi(_ pi: String, _ favoriteNumbers: [String]) -> Int {
6         var numbersDictionary = [String: Bool]()
7
8         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) {
15            getMinimumNumberOfSpaces(pi, numbersDictionary, &cache, i)
16        }
17
18        if cache[0] == Int(Int32.max) {
19            return -1
20        } else {
21            return cache[0]!
22        }
23    }
24
25    func getMinimumNumberOfSpaces(_ pi: String, _ numbersDictionary: [String: Bool], _ cache: [Int: Int]) {
26        if index == pi.count {
27            return -1
28        }
29
30        if let minimumNumberOfSpaces = cache[index] {
31            return minimumNumberOfSpaces
32        }
33    }
34}
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^3 + m) time | O(n + m) space
5     func numbersInPi(_ pi: String, _ favoriteNumbers: [String]) -> Int {
6         // Write your code here.
7         return -1
8     }
9 }
```

```

18         assertAlmostEqual(2, integral.quad(func, firstBound, secondBound))
19
20     def test_quad_2D(self):
21         func = (lambda x, y: 2*x*y, 'xy', 'xy', 'xy', 'xy', 'xy')
22         assertAlmostEqual(integral.quad(func, firstBound, secondBound),
23                             integral.quad(func, firstBound, secondBound))
24         assertAlmostEqual(integral.quad(func, firstBound, secondBound),
25                             integral.quad(func, firstBound, secondBound))
26
27     def test_quad_3D(self):
28         func = (lambda x, y, z: 2*x*y*z, 'xyz', 'xyz', 'xyz', 'xyz', 'xyz')
29         assertAlmostEqual(integral.quad(func, firstBound, secondBound),
30                             integral.quad(func, firstBound, secondBound))
31         assertAlmostEqual(integral.quad(func, firstBound, secondBound),
32                             integral.quad(func, firstBound, secondBound))
33
34     def test_quad_4D(self):
35         func = (lambda x, y, z, w: 2*x*y*z*w, 'xyzw', 'xyzw', 'xyzw', 'xyzw', 'xyzw')
36         assertAlmostEqual(integral.quad(func, firstBound, secondBound),
37                             integral.quad(func, firstBound, secondBound))

```

Run or submit code when you're ready.