```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  using System;
4  using System.Collections.Generic;
5
6  public class Program {
7    // O(n^3 + m) time | O(n + m) space - where n is the number of digits in Pi and m is the number of favorite numbers
8    public static int NumbersInPi(string pi, string[] numbers) {
9      HashSet<string> numbersTable = new HashSet<string>();
10     foreach (string number in numbers) {
11       numbersTable.Add(number);
12     }
13     Dictionary<int, int> cache = new Dictionary<int, int>();
14     int minSpaces = getMinSpaces(pi, numbersTable, cache, 0);
15     return minSpaces == Int32.MaxValue ? -1 : minSpaces;
16   }
17
18   public static int getMinSpaces(
19     string pi,
20     HashSet<string> numbersTable,
21     Dictionary<int, int> cache,
22     int idx
23     ) {
24     if (idx == pi.Length) return -1;
25     if (cache.ContainsKey(idx)) return cache[idx];
26     int minSpaces = Int32.MaxValue;
27     for (int i = idx; i < pi.Length; i++) {
28       string prefix = pi.Substring(idx, i + 1 - idx);
29       if (numbersTable.Contains(prefix)) {
30         int minSpacesInSuffix =
31           getMinSpaces(pi, numbersTable, cache, i + 1);
32         // Handle int overflow.
33         if (minSpacesInSuffix == Int32.MaxValue) {
34           minSpaces = Math.Min(minSpaces, minSpacesInSuffix);
35         } else {
36           minSpaces = Math.Min(minSpaces, minSpacesInSuffix + 1);
37         }
38       }
39     }
40     cache[idx] = minSpaces;
41     return cache[idx];
42   }
43 }
44
```