

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(n^3 + m) time | O(n + m) space - where n is the number of digits
4 def numbersInPi(pi, numbers):
5     numbersTable = {number: True for number in numbers}
6     cache = {}
7     for i in reversed(range(len(pi))):
8         getMinSpaces(pi, numbersTable, cache, i)
9     return -1 if cache[0] == float("inf") else cache[0]
10
11
12 def getMinSpaces(pi, numbersTable, cache, idx):
13     if idx == len(pi):
14         return -1
15     if idx in cache:
16         return cache[idx]
17     minSpaces = float("inf")
18     for i in range(idx, len(pi)):
19         prefix = pi[idx : i + 1]
20         if prefix in numbersTable:
21             minSpacesInSuffix = getMinSpaces(pi, numbersTable, cache,
22                                               i + 1)
23             minSpaces = min(minSpaces, minSpacesInSuffix + 1)
24     cache[idx] = minSpaces
25     return cache[idx]
```

Solution 1Solution 2Solution 3

```
1 def numbersInPi(pi, numbers):
2     # Write your code here.
3     pass
4
```

Our Tests

Custom Output

Submit Code

```

1  #!/usr/bin/env python
2
3  # Import the random module
4  import random
5
6  # Create a list of numbers
7  numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
8
9  # Shuffle the list
10 random.shuffle(numbers)
11
12 # Print the shuffled list
13 print(numbers)
14
15 # Create a list of names
16 names = ["John", "Jane", "Bob", "Alice", "Charlie", "David", "Eve", "Frank"]
17
18 # Shuffle the list
19 random.shuffle(names)
20
21 # Print the shuffled list
22 print(names)

```

Run or submit code when you're ready.