

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 import "math"
6
7 // O(n^3 + m) time | O(n + m) space - where n is the number of digits
8 // in Pi and m is the number of favorite numbers.
9 func NumbersInPi(pi string, numbers []string) int {
10     numbersTable := map[string]bool{}
11     for _, number := range numbers {
12         numbersTable[number] = true
13     }
14
15     cache := map[int]int{}
16     for i := len(pi) - 1; i >= 0; i-- {
17         getMinSpaces(pi, numbersTable, cache, i)
18     }
19
20     if cache[0] == math.MaxInt32 {
21         return -1
22     }
23     return cache[0]
24 }
25
26 func getMinSpaces(pi string, numbersTable map[string]bool,
27     cache map[int]int, idx int) int {
28     if idx == len(pi) {
29         return -1
30     } else if val, found := cache[idx]; found {
31         return val
32     }
33     minSpaces := math.MaxInt32
```

Solution 1Solution 2Solution 3

```
1 package main
2
3 func NumbersInPi(pi string, numbers []string) int {
4     // Write your code here.
5     return -1
6 }
7
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

```
10 return 0
11
12 def isPrime(n):
13     return isPrimeRec(n, 2)
14
15 def isPrimeRec(n, p):
16     if p >= n:
17         return True
18     if n % p == 0:
19         return False
20     return isPrimeRec(n, p + 1)
21
22 def isPrime(n):
23     return isPrimeRec(n, 2)
24
25 def isPrime(n):
26     return isPrimeRec(n, 2)
27
28 def isPrime(n):
29     return isPrimeRec(n, 2)
```

Run or submit code when you're ready.