```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# 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
def numbersInPi(pi, numbers):
    numbersTable = {number: True for number in numbers}
    cache = {}
    for i in reversed(range(len(pi))):
        getMinSpaces(pi, numbersTable, cache, i)
    return -1 if cache[0] == float("inf") else cache[0]


def getMinSpaces(pi, numbersTable, cache, idx):
    if idx == len(pi):
        return -1
    if idx in cache:
        return cache[idx]
    minSpaces = float("inf")
    for i in range(idx, len(pi)):
        prefix = pi[idx : i + 1]
        if prefix in numbersTable:
            minSpacesInSuffix = getMinSpaces(pi, numbersTable, cache, i + 1)
            minSpaces = min(minSpaces, minSpacesInSuffix + 1)
    cache[idx] = minSpaces
    return cache[idx]

```