Our Solution(s) Run Code

Solution 1 Solution 2

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// O(n^3 + m) time | O(n + m) space - where n is the number of digits
function numbersInPi(pi, numbers) {
  const numbersTable = {};
  for (const number of numbers) {
    numbersTable[number] = true;
  }
  const cache = {};
  for (let i = pi.length - 1; i >= 0; i--) {
    getMinSpaces(pi, numbersTable, cache, i);
  }
  return cache[0] === Infinity ? -1 : cache[0];
}

function getMinSpaces(pi, numbersTable, cache, idx) {
  if (idx === pi.length) return -1;
  if (idx in cache) return cache[idx];
  let minSpaces = Infinity;
  for (let i = idx; i < pi.length; i++) {
    const prefix = pi.slice(idx, i + 1);
    if (prefix in numbersTable) {
      const minSpacesInSuffix = getMinSpaces(pi, numbersTable, cache,
      minSpaces = Math.min(minSpaces, minSpacesInSuffix + 1);
    }
  }
  cache[idx] = minSpaces;
  return cache[idx];
}

exports.numbersInPi = numbersInPi;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function numbersInPi(pi, numbers) {
  // Write your code here.
}

// Do not edit the line below.
exports.numbersInPi = numbersInPi;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.