```
// 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
function numbersInPi(pi, numbers) {
  const numbersTable = {};
  for (const number of numbers) {
    numbersTable[number] = true;
  }
  const minSpaces = getMinSpaces(pi, numbersTable, {}, 0);
  return minSpaces === Infinity ? -1 : minSpaces;
}

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, i + 1);
      minSpaces = Math.min(minSpaces, minSpacesInSuffix + 1);
    }
  }
  cache[idx] = minSpaces;
  return cache[idx];
}

exports.numbersInPi = numbersInPi;

```