AlgoExpert Quad Layout Java 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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

import java.util.*;

class Program {
  // 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
  public static int numbersInPi(String pi, String[] numbers) {
    Set<String> numbersTable = new HashSet<String>();
    for (String number : numbers) {
      numbersTable.add(number);
    }
    Map<Integer, Integer> cache = new HashMap<Integer, Integer>();
    for (int i = pi.length() - 1; i >= 0; i--) {
      getMinSpaces(pi, numbersTable, cache, i);
    }
    return cache.get(0) == Integer.MAX_VALUE ? -1 : cache.get(0);
  }

  public static int getMinSpaces(
      String pi, Set<String> numbersTable, Map<Integer, Integer> cache, int idx) {
    if (idx == pi.length()) return -1;
    if (cache.containsKey(idx)) return cache.get(idx);
    int minSpaces = Integer.MAX_VALUE;
    for (int i = idx; i < pi.length(); i++) {
      String prefix = pi.substring(idx, i + 1);
      if (numbersTable.contains(prefix)) {
        int minSpacesInSuffix = getMinSpaces(pi, numbersTable, cache, i + 1);
        // Handle int overflow.
        if (minSpacesInSuffix == Integer.MAX_VALUE) {
          minSpaces = Math.min(minSpaces, minSpacesInSuffix);
        } else {
          minSpaces = Math.min(minSpaces, minSpacesInSuffix + 1);
        }
      }
    }
    cache.put(idx, minSpaces);
    return cache.get(idx);
  }
}
```