AlgoExpert Quad Layout C++ 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

48 }

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 #include <set>
 4 #include <unordered_map>
 5 #include <algorithm>
 6 #include <climits>
 7 #include <vector>
8 using namespace std;
10 int getMinSpaces(string pi, set<string> numbersTable,
                   unordered_map<int, int> *cache, int idx);
11
12
13 \ //\ O(n^3+m) time \ |\ O(n+m) space - where n is the number of digits in Pi and
14 // m is the number of favorite numbers
15 int numbersInPi(string pi, vector<string> numbers) {
    set<string> numbersTable;
16
17
      for (string number : numbers) {
18
       numbersTable.insert(number);
19
20
     unordered_map<int, int> cache;
21
      for (int i = pi.length() - 1; i >= 0; i--) {
22
       getMinSpaces(pi, numbersTable, &cache, i);
23
24
     return cache.at(0) == INT_MAX ? -1 : cache.at(0);
25 }
26
27 int getMinSpaces(string pi, set<string> numbersTable,
28
                    unordered_map<int, int> *cache, int idx) {
     if (idx == pi.length())
29
       return -1;
30
      if (cache->find(idx) != cache->end())
31
       return cache->at(idx);
32
      int minSpaces = INT_MAX;
33
      for (int i = idx; i < pi.length(); i++) {</pre>
34
       string prefix = pi.substr(idx, i + 1 - idx);
35
       if (numbersTable.find(prefix) != numbersTable.end()) {
36
37
         int minSpacesInSuffix = getMinSpaces(pi, numbersTable, cache, i + 1);
38
          // Handle int overflow.
          if (minSpacesInSuffix == INT_MAX) {
39
           minSpaces = min(minSpaces, minSpacesInSuffix);
40
41
          } else {
           minSpaces = min(minSpaces, minSpacesInSuffix + 1);
42
43
44
45
     cache->insert({idx, minSpaces});
46
      return cache->at(idx);
47
```