

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <algorithm>
4 #include <vector>
5 #include <numeric>
6 using namespace std;
7
8 vector<int> getLocalMinIdxs(vector<int> array);
9 void expandFromLocalMinIdx(int localMinIdx, vector<int> scores,
10                             vector<int> *rewards);
11
12 // O(n) time | O(n) space - where n is the length of the input array
13 int minRewards(vector<int> scores) {
14     vector<int> rewards = vector<int>(scores.size(), 1);
15     vector<int> localMinIdxs = getLocalMinIdxs(scores);
16     for (int localMinIdx : localMinIdxs) {
17         expandFromLocalMinIdx(localMinIdx, scores, &rewards);
18     }
19     return accumulate(rewards.begin(), rewards.end(), 0);
20 }
21
22 vector<int> getLocalMinIdxs(vector<int> array) {
23     if (array.size() == 1)
24         return vector<int>{0};
25     vector<int> localMinIdxs = {};
26     for (int i = 0; i < array.size(); i++) {
27         if (i == 0 && array[i] < array[i + 1])
28             localMinIdxs.push_back(i);
29         if (i == array.size() - 1 && array[i] < array[i - 1])
30             localMinIdxs.push_back(i);
31         if (i == 0 || i == array.size() - 1)
32             continue;
33         if (array[i] < array[i + 1] && array[i] < array[i - 1])
34             localMinIdxs.push_back(i);
35     }
36     return localMinIdxs;
37 }
38
39 void expandFromLocalMinIdx(int localMinIdx, vector<int> scores,
40                             vector<int> *rewards) {
41     int leftIdx = localMinIdx - 1;
42     while (leftIdx >= 0 && scores[leftIdx] > scores[leftIdx + 1]) {
43         rewards->at(leftIdx) =
44             max(rewards->at(leftIdx), rewards->at(leftIdx + 1) + 1);
45         leftIdx--;
46     }
47     int rightIdx = localMinIdx + 1;
48     while (rightIdx < scores.size() && scores[rightIdx] > scores[rightIdx - 1]) {
49         rewards->at(rightIdx) = rewards->at(rightIdx - 1) + 1;
50         rightIdx++;
51     }
52 }
```