Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3

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

using System;
using System.Linq;
using System.Collections.Generic;

// O(n) time | O(n) space - where in is the length of the input array
public class Program {
  public static int MinRewards(int[] scores) {
    int[] rewards = new int[scores.Length];
    Array.Fill(rewards, 1);
    List<int> localMinIdxs = getLocalMinIdxs(scores);
    foreach (int localMinIdx in localMinIdxs) {
      expandFromLocalMinIdx(localMinIdx, scores, rewards);
    }
    return rewards.Sum();
  }

  public static List<int> getLocalMinIdxs(int[] array) {
    List<int> localMinIdxs = new List<int>();
    if (array.Length == 1) {
      localMinIdxs.Add(0);
      return localMinIdxs;
    }
    for (int i = 0; i < array.Length; i++) {
      if (i == 0 && array[i] < array[i + 1]) localMinIdxs.Add(i);
      if (i == array.Length - 1 && array[i] < array[i - 1]) localMinId
      if (i == 0 || i == array.Length - 1) continue;
      if (array[i] < array[i + 1] && array[i] < array[i - 1]) localMin
    }
    return localMinIdxs;
  }

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
public class Program {
  public static int MinRewards(int[] scores) {
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

```
}

[Test]
public void TestCase01() {
  var input = new int[] {1, 10};
  var actual = Program.MinMoves(input);
  var expected = 1;
  Utils.AssertEqual(expected, actual);
}

[Test]
public void TestCase02() {
  var input = new int[] {10, 1};
  var actual = Program.MinMoves(input);
  var expected = 1;
```

Run or submit code when you're ready.