

Our Solution(s) Run Code

Our Solution(s) Run Code

Solution 1	Solution 2	Solution 3
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.		
2		
3 package main		
4		
5 // O(n) time O(n) space - where in is the length of the input array		
6 func MinRewards(scores []int) int {		
7 rewards := make([]int, len(scores))		
8 fill(rewards, 1)		
9 for i := 1; i < len(scores); i++ {		
10 if scores[i] > scores[i-1] {		
11 rewards[i] = rewards[i-1] + 1		
12 }		
13 }		
14 for i := len(scores) - 2; i >= 0; i-- {		
15 if scores[i] > scores[i+1] {		
16 rewards[i] = max(rewards[i], rewards[i+1]+1)		
17 }		
18 }		
19 return sum(rewards)		
20 }		
21		
22 func fill(arr []int, val int) {		
23 for i := range arr {		
24 arr[i] = val		
25 }		
26 }		
27		
28 func sum(arr []int) int {		
29 sum := 0		
30 for i := range arr {		
31 sum += arr[i]		
32 }		
33 return sum		

Your Solutions

Run Code

Your Solutions

Run Code

Our Tests

Custom Output

Submit Code

Custom Output

Submit Code

```
14 from typing import TypedDict, List
15 input = TypedDict(
16     output = List[int]
17     expected = 5
18     expected_output, expected, output
19 )
20
21 from typing import TypedDict, List
22 input = TypedDict(
23     output = List[int]
24     expected = 5
25     expected_output, expected, output
26 )
27
```

Run or submit code when you're ready.