

Our Solution(s)

Run Code

Run Code

Solution 1	Solution 2	Solution 3
------------	------------	------------

```

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(n^2) time | O(n) space - where n 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         j := i - 1
11         if scores[i] > scores[j] {
12             rewards[i] = rewards[j] + 1
13             continue
14         }
15         for j >= 0 && scores[j] > scores[j+1] {
16             rewards[j] = max(rewards[j], rewards[j+1]+1)
17             j--
18         }
19     }
20     return sum(rewards)
21 }
22
23 func fill(arr []int, val int) {
24     for i := range arr {
25         arr[i] = val
26     }
27 }
28
29 func sum(arr []int) int {
30     sum := 0
31     for i := range arr {
32         sum += arr[i]
33     }

```

Your Solutions

Run Code

Run Code

Solution 1	Solution 2	Solution 3
------------	------------	------------

```
1 package main
2
3 func MinRewards(scores []int) int {
4     // Write your code here.
5     return -1
6 }
7
```

Our Tests

Custom Output

Submit Code

Submit Code

```
14 def main():
15     input = input()
16     output = solve(input)
17     print(output)
18
19 if __name__ == '__main__':
20     main()
21
22 def main():
23     input = input()
24     output = solve(input)
25     print(output)
26
27 if __name__ == '__main__':
28     main()
29
```

Run or submit code when you're ready.