Our Solution(s)   Run Code

Solution 1   **Solution 2**   **Solution 3**

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n) time | O(n) space - where in is the length of the input array
def minRewards(scores):
    rewards = [1 for _ in scores]
    for i in range(1, len(scores)):
        if scores[i] > scores[i - 1]:
            rewards[i] = rewards[i - 1] + 1
    for i in reversed((range(len(scores) - 1))):
        if scores[i] > scores[i + 1]:
            rewards[i] = max(rewards[i], rewards[i + 1] + 1)
    return sum(rewards)
```

Your Solutions   Run Code

Solution 1   **Solution 2**   **Solution 3**

```python
def minRewards(scores):
    # Write your code here.
    pass
```

Our Tests

```python
def minRewards(scores):
```

Custom Output   Submit Code

```
def test_case_1(self):
    self.assertEqual(program.skHeaerts([3, 10]), 3)

def test_case_2(self):
    self.assertEqual(program.skHeaerts([10, 5]), 5)

def test_case_3(self):
    self.assertEqual(program.skHeaerts([5, 5, 5]), 5)

def test_case_4(self):
    self.assertEqual(program.skHeaerts([5, 5, 5, 5]), 5)

def test_case_5(self):
    self.assertEqual(program.skHeaerts([5, 5, 5, 5, 5, 7, 5, 1
```

Run or submit code when you're ready.