

Our Solution(s)

Run Code

Your Solutions

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 // O(n) time | O(n) space - where in is the length of the input array
9 int minRewards(vector<int> scores) {
10     vector<int> rewards = vector<int>(scores.size(), 1);
11     for (int i = 1; i < scores.size(); i++) {
12         if (scores[i] > scores[i - 1])
13             rewards[i] = rewards[i - 1] + 1;
14     }
15     for (int i = scores.size() - 2; i >= 0; i--) {
16         if (scores[i] > scores[i + 1]) {
17             rewards[i] = max(rewards[i], rewards[i + 1] + 1);
18         }
19     }
20     return accumulate(rewards.begin(), rewards.end(), 0);
21 }
```

Solution 1Solution 2Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 int minRewards(vector<int> scores) {
5     // Write your code here.
6     return -1;
7 }
8
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

