

```
class Solution {
public:
    int candy(vector<int>& ratings) {
        int array_length = ratings.size(), total_count = 0;
        vector<int> counters(array_length, 1);

        if (array_length == 1) {
            return 1;
        }

        for (int index = 0; index < array_length - 1; index++) {
            if (ratings[index + 1] > ratings[index]) {
                counters[index + 1] = counters[index] + 1;
            }
        }

        for (int index = array_length - 2; index >= 0; index--) {
            if (ratings[index] > ratings[index + 1] && counters[index] <= counters[index + 1]) {
                counters[index] = counters[index + 1] + 1;
            }
            total_count += counters[index];
        }
        total_count += counters[array_length - 1];

        return total_count;
    }
};
```

Greedy

DescriptionAcceptedEditorialSolutionsSubmissions

135. Candy

Solved

HardTopicsCompanies

There are n children standing in a line. Each child is assigned a rating value given in the integer array `ratings`.

You are giving candies to these children subjected to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.

Return the minimum number of candies you need to have to distribute the candies to the children.

Example 1:

Input: `ratings = [1,0,2]`
Output: 5
Explanation: You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

Example 2:

Input: `ratings = [1,2,2]`
Output: 4
Explanation: You can allocate to the first, second and third child with 1, 2, 1 candies respectively. The third child gets 1 candy because it satisfies the above two conditions.

Constraints:

- $n == ratings.length$
- $1 \leq n \leq 2 \cdot 10^5$
- $0 \leq ratings[i] \leq 2 \cdot 10^5$

Seen this question in a real interview before? 1/5

Yes No

8.2K 267 81 Online

Code

C++Auto

```
1 class Solution {
2 public:
3     candy(vector<int>& ratings) {
4         int array_length = ratings.size(), total_count = 0;
5         vector<int> counters(array_length, 1);
6
7         if (array_length == 1) {
8             return 1;
9         }
10
11         for (int index = 0; index < array_length - 1; index++) {
12             if (ratings[index + 1] > ratings[index]) {
13                 counters[index + 1] = counters[index] + 1;
14             }
15         }
16
17         for (int index = array_length - 2; index >= 0; index--) {
18             if (ratings[index] > ratings[index + 1] && counters[index] <= counters[index + 1]) {
19                 counters[index] = counters[index + 1] + 1;
20             }
21         }
22
23         for (int i = 0; i < array_length; i++) {
24             total_count += counters[i];
25         }
26
27         return total_count;
28     }
29 }
```

Ln 10, Col 1 Saved

Run Submit

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

ratings =
[1,0,2]

Output

5

Expected

5

Contribute a testcase

Greedy

DescriptionAcceptedEditorialSolutionsSubmissions

135. Candy

Solved

HardTopicsCompanies

There are n children standing in a line. Each child is assigned a rating value given in the integer array `ratings`.

You are giving candies to these children subjected to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.

Return the minimum number of candies you need to have to distribute the candies to the children.

Example 1:

Input: `ratings = [1,0,2]`
Output: 5
Explanation: You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

Example 2:

Input: `ratings = [1,2,2]`
Output: 4
Explanation: You can allocate to the first, second and third child with 1, 2, 1 candies respectively. The third child gets 1 candy because it satisfies the above two conditions.

Constraints:

- $n == ratings.length$
- $1 \leq n \leq 2 \cdot 10^5$
- $0 \leq ratings[i] \leq 2 \cdot 10^5$

Seen this question in a real interview before? 1/5

Yes No

8.2K 267 81 Online

Code

C++Auto

```
1 class Solution {
2 public:
3     candy(vector<int>& ratings) {
4         int array_length = ratings.size(), total_count = 0;
5         vector<int> counters(array_length, 1);
6
7         if (array_length == 1) {
8             return 1;
9         }
10
11         for (int index = 0; index < array_length - 1; index++) {
12             if (ratings[index + 1] > ratings[index]) {
13                 counters[index + 1] = counters[index] + 1;
14             }
15         }
16
17         for (int index = array_length - 2; index >= 0; index--) {
18             if (ratings[index] > ratings[index + 1] && counters[index] <= counters[index + 1]) {
19                 counters[index] = counters[index + 1] + 1;
20             }
21         }
22
23         for (int i = 0; i < array_length; i++) {
24             total_count += counters[i];
25         }
26
27         return total_count;
28     }
29 }
```

Ln 10, Col 1 Saved

Run Submit

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

ratings =
[1,2,2]

Output

4

Expected

4

Contribute a testcase

Асимптотика этого алгоритма — $O(n)$, где n — размер вектора ratings. Это связано с тем, что алгоритм выполняет два вложенных цикла, каждый из которых проходит по всему вектору (от 0 до $n-1$ и от $n-2$ до 0). Таким образом, общее количество операций пропорционально размеру вектора.

