

0135. 分发糖果

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：贪心、数组
- 难度：困难

题目链接

- [0135. 分发糖果 - 力扣](#)

题目大意

描述： n 个孩子站成一排。老师会根据每个孩子的表现，给每个孩子进行评分。然后根据下面的规则给孩子们分发糖果：

- 每个孩子至少得 1 个糖果。
- 评分更高的孩子必须比他两侧相邻位置上的孩子分得更多的糖果。

现在给定 n 个孩子的表现分数数组 `ratings`，其中 `ratings[i]` 表示第 i 个孩子的评分。

要求： 返回最少需要准备的糖果数目。

说明：

- $n == ratings.length$ 。
- $1 \leq n \leq 2 \times 10^4$ 。
- $0 \leq ratings[i] \leq 2 * 10^4$ 。

示例：

- 示例 1:

输入: `ratings = [1,0,2]`

输出: `5`

解释: 你可以分别给第一个、第二个、第三个孩子分发 `2`、`1`、`2` 颗糖果。

py

- 示例 2:

输入: ratings = [1,2,2]

输出: 4

解释: 你可以分别给第一个、第二个、第三个孩子分发 1、2、1 颗糖果。

第三个孩子只得到 1 颗糖果, 这满足题面中的两个条件。

解题思路

思路 1: 贪心算法

先来看分发糖果的规则。

「每个孩子至少得 1 个糖果」: 说明糖果数目至少为 N 个。

「评分更高的孩子必须比他两侧相邻位置上的孩子分得更多的糖果」: 可以看做为以下两种条件:

- 当 $ratings[i - 1] < ratings[i]$ 时, 第 i 个孩子的糖果数量比第 $i - 1$ 个孩子的糖果数量多;
- 当 $ratings[i] > ratings[i + 1]$ 时, 第 i 个孩子的糖果数量比第 $i + 1$ 个孩子的糖果数量多。

根据以上信息, 我们可以设定一个长度为 N 的数组 `sweets` 来表示每个孩子分得的最少糖果数, 初始每个孩子分得糖果数都为 1。

然后遍历两遍数组, 第一遍遍历满足当 $ratings[i - 1] < ratings[i]$ 时, 第 i 个孩子的糖果数量比第 $i - 1$ 个孩子的糖果数量多 1 个。第二遍遍历满足当 $ratings[i] > ratings[i + 1]$ 时, 第 i 个孩子的糖果数量取「第 $i + 1$ 个孩子的糖果数量多 1 个」和「第 $i + 1$ 个孩子目前拥有的糖果数量」中的最大值。

然后再遍历求所有孩子的糖果数量和即为答案。

思路 1: 代码

```
class Solution:
    def candy(self, ratings: List[int]) -> int:
        size = len(ratings)
        sweets = [1 for _ in range(size)]

        for i in range(1, size):
```

```
    if ratings[i] > ratings[i - 1]:
        sweets[i] = sweets[i - 1] + 1

for i in range(size - 2, -1, -1):
    if ratings[i] > ratings[i + 1]:
        sweets[i] = max(sweets[i], sweets[i + 1] + 1)

res = sum(sweets)
return res
```

思路 1：复杂度分析

- 时间复杂度： $O(n)$ ，其中 n 是数组 `ratings` 的长度。
- 空间复杂度： $O(n)$ 。