135. Candy (Hard)

一群孩子站成一排，每一个孩子有自己的评分。现在需要给这些孩子发糖果，规则是如果一 个孩子的评分比自己身旁的一个孩子要高，那么这个孩子就必须得到比身旁孩子更多的糖果;所有孩子至少要有一个糖果。求解最少需要多少个糖果。

输入是一个数组，表示孩子的评分。输出是最少糖果的数量。
Input: [1,0,2] Output: 5 在这个样例中，最少的糖果分法是 [2,1,2]

做完了题目 455 ,你会不会认为存在比较关系的贪心策略一定需要排序或是选择 ？
虽然这一 道题也是运用贪心策略，但我们只需要简单的两次遍历即可:把所有孩子的糖果数初始化为 1; 先从左往右遍历一遍，如果右边孩子的评分比左边的高，则右边孩子的糖果数更新为左边孩子的 糖果数加 1;再从右往左遍历一遍，如果左边孩子的评分比右边的高，且左边孩子当前的糖果数 不大于右边孩子的糖果数，则左边孩子的糖果数更新为右边孩子的糖果数加 1。通过这两次遍历， 分配的糖果就可以满足题目要求了。这里的贪心策略即为，在每次遍历中，只考虑并更新相邻一 侧的大小关系。

```
 var candy = function(ratings) {
       let N = ratings.length;
       if (N < 2) return N;
       //把所有孩子的糖果数初始化为 1
       let res = Array(N).fill(1);
       // 先从左往右遍历一遍，如果右边孩子的评分比左边的高，则右边孩子的糖果数更新为左边孩子的 糖果数加 1;
       for (let i = 1; i < N; i++) {
           if (ratings[i] > ratings[i - 1]) {
               res[i] = res[i - 1] + 1;
           }
       }
       // 再从右往左遍历一遍，如果左边孩子的评分比右边的高，且左边孩子当前的糖果数不大于右边孩子的糖果数，则左边孩子的糖果数更新为右边孩子的糖果数加 1
       for (let i = N - 1; i > 0; i--) {
           if (ratings[i] < ratings[i - 1]) {
               res[i - 1] = Math.max(res[i - 1], res[i] + 1);;
           }
       }
       //求和
       return res.reduce((a, b) => a + b, 0);
   };
```