

1402-做菜顺序

题述

1402. 做菜顺序

难度 困难 94 ☆ 17 20 20 20 20

一个厨师收集了他 n 道菜的满意程度 $satisfaction$ ，这个厨师做出每道菜的时间都是 1 单位时间。

一道菜的「喜爱时间」系数定义为烹饪这道菜以及之前每道菜所花费的时间乘以这道菜的满意程度，也就是 $time[i] * satisfaction[i]$ 。

请你返回做完所有菜「喜爱时间」总和的最大值为多少。

你可以按任意顺序安排做菜的顺序，你也可以选择放弃做某些菜来获得更大的总和。

示例 1:

输入: $satisfaction = [-1,-8,0,5,-9]$

输出: 14

解释: 去掉第二道和最后一道菜，最大的喜爱时间系数和为 $(-1*1 + 0*2 + 5*3 = 14)$ 。每道菜都需要花费 1 单位时间完成。

示例 2:

输入: $satisfaction = [4,3,2]$

输出: 20

解释: 按照原来顺序相反的时间做菜 $(2*1 + 3*2 + 4*3 = 20)$

示例 3:

输入: $satisfaction = [-1,-4,-5]$

输出: 0

解释: 大家都不喜欢这些菜，所以不做任何菜可以获得最大的喜爱时间系数。

思路

这个题目

观察示例 得出公式

时间 $time[i] * satisfaction[i]$

发现贪心策略: 满意程度越高的菜应该放在越后面

注意：题目有提到，可以放弃某些菜，即可以放弃满意程度较低的菜

考虑使用排序

- 将所有菜按照满意程度排序
- 按照排好序的顺序遍历序列，对于当前遍历到的菜，如果它与之前选择的所有菜的满意程度之和大于 0，我们就选择这道菜，否则可以直接退出遍历的循环。
- 这是因为如果 sis_isi 与之前选择的所有菜的满意程度之和已经小于等于 0 了，那么后面的菜比 si 的满意程度还要小，就更不可能得到一个大于 0 的和了。

题解

Python贪心

```
class Solution:
    def maxSatisfaction(self, satisfaction: List[int]) -> int:
        satisfaction.sort()
        result = 0 #记录最终的结果
        a=0
        b=0
        for i in range(len(satisfaction)-1,-1,-1):
            #从后往前遍历
            a += satisfaction[i]
            b += a
            result = max(result,b) #贪心的体现
        return result
```

思考

思考贪心策略

这题竟然是Hard