Problem 1402: Reducing Dishes

Problem Information

Difficulty: Hard
Acceptance Rate: 76.51%
Paid Only: No
Tags: Array, Dynamic Programming, Greedy, Sorting

Problem Description

A chef has collected data on the `satisfaction` level of his `n` dishes. Chef can cook any dish in 1 unit of time.

Like-time coefficient of a dish is defined as the time taken to cook that dish including previous dishes multiplied by its satisfaction level i.e. `time[i] * satisfaction[i]`.

Return the maximum sum of **like-time coefficient** that the chef can obtain after preparing some amount of dishes.

Dishes can be prepared in **any** order and the chef can discard some dishes to get this maximum value.

Example 1:

Input: satisfaction = [-1,-8,0,5,-9] **Output:** 14 **Explanation:** After Removing the second and last dish, the maximum total **like-time coefficient** will be equal to (-1*1 + 0*2 + 5*3 = 14). Each dish is prepared in one unit of time.

Example 2:

Input: satisfaction = [4,3,2] **Output:** 20 **Explanation:** Dishes can be prepared in any order, (2*1 + 3*2 + 4*3 = 20)

Example 3:

Input: satisfaction = [-1,-4,-5] **Output:** 0 **Explanation:** People do not like the dishes. No dish is prepared.

Constraints:

* `n == satisfaction.length` * `1 <= n <= 500` * `-1000 <= satisfaction[i] <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int maxSatisfaction(vector<int>& satisfaction) {

}
};
```

Java:

```
class Solution {
public int maxSatisfaction(int[] satisfaction) {

}
}
```

Python3:

```
class Solution:
def maxSatisfaction(self, satisfaction: List[int]) -> int:
```