Problem 2144: Minimum Cost of Buying Candies With Discount

Problem Information

Difficulty: Easy
Acceptance Rate: 62.56%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

A shop is selling candies at a discount. For **every two** candies sold, the shop gives a **third** candy for **free**.

The customer can choose **any** candy to take away for free as long as the cost of the chosen candy is less than or equal to the **minimum** cost of the two candies bought.

* For example, if there are `4` candies with costs `1`, `2`, `3`, and `4`, and the customer buys candies with costs `2` and `3`, they can take the candy with cost `1` for free, but not the candy with cost `4`.

Given a **0-indexed** integer array `cost`, where `cost[i]` denotes the cost of the `ith` candy, return _the**minimum cost** of buying **all** the candies_.

Example 1:

Input: cost = [1,2,3] **Output:** 5 **Explanation:** We buy the candies with costs 2 and 3, and take the candy with cost 1 for free. The total cost of buying all candies is 2 + 3 = 5. This is the **only** way we can buy the candies. Note that we cannot buy candies with costs 1 and 3, and then take the candy with cost 2 for free. The cost of the free candy has to be less than or equal to the minimum cost of the purchased candies.

Example 2:

Input: cost = [6,5,7,9,2,2] **Output:** 23 **Explanation:** The way in which we can get the minimum cost is described below: - Buy candies with costs 9 and 7 - Take the candy with cost

6 for free - We buy candies with costs 5 and 2 - Take the last remaining candy with cost 2 for free Hence, the minimum cost to buy all candies is 9 + 7 + 5 + 2 = 23.

Example 3:

Input: cost = [5,5] **Output:** 10 **Explanation:** Since there are only 2 candies, we buy both of them. There is not a third candy we can take for free. Hence, the minimum cost to buy all candies is 5 + 5 = 10.

Constraints:

* `1 <= cost.length <= 100` * `1 <= cost[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int minimumCost(vector<int>& cost) {

}
};
```

Java:

```
class Solution {
public int minimumCost(int[] cost) {

}
}
```

Python3:

```
class Solution:
def minimumCost(self, cost: List[int]) -> int:
```