Problem 2561: Rearranging Fruits

Problem Information

Difficulty: Hard
Acceptance Rate: 57.58%
Paid Only: No
Tags: Array, Hash Table, Greedy, Sort

Problem Description

You have two fruit baskets containing `n` fruits each. You are given two **0-indexed** integer arrays `basket1` and `basket2` representing the cost of fruit in each basket. You want to make both baskets **equal**. To do so, you can use the following operation as many times as you want:

* Choose two indices `i` and `j`, and swap the `ith` fruit of `basket1` with the `jth` fruit of `basket2`. * The cost of the swap is `min(basket1[i], basket2[j])`.

Two baskets are considered equal if sorting them according to the fruit cost makes them exactly the same baskets.

Return _the minimum cost to make both the baskets equal or_`-1` _if impossible._

Example 1:

Input: basket1 = [4,2,2,2], basket2 = [1,4,1,2] **Output:** 1 **Explanation:** Swap index 1 of basket1 with index 0 of basket2, which has cost 1. Now basket1 = [4,1,2,2] and basket2 = [2,4,1,2]. Rearranging both the arrays makes them equal.

Example 2:

Input: basket1 = [2,3,4,1], basket2 = [3,2,5,1] **Output:** -1 **Explanation:** It can be shown that it is impossible to make both the baskets equal.

Constraints:

* `basket1.length == basket2.length` * `1 <= basket1.length <= 105` * `1 <= basket1[i], basket2[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long minCost(vector<int>& basket1, vector<int>& basket2) {

}
};
```

Java:

```
class Solution {
public long minCost(int[] basket1, int[] basket2) {

}
}
```

Python3:

```
class Solution:
def minCost(self, basket1: List[int], basket2: List[int]) -> int:
```