Problem 1029: Two City Scheduling

Problem Information

Difficulty: Medium
Acceptance Rate: 68.11%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

A company is planning to interview `2n` people. Given the array `costs` where `costs[i] = [aCosti, bCosti]`, the cost of flying the `ith` person to city `a` is `aCosti`, and the cost of flying the `ith` person to city `b` is `bCosti`.

Return _the minimum cost to fly every person to a city_ such that exactly `n` people arrive in each city.

Example 1:

Input: costs = [[10,20],[30,200],[400,50],[30,20]] **Output:** 110 **Explanation:** The first person goes to city A for a cost of 10. The second person goes to city A for a cost of 30. The third person goes to city B for a cost of 50. The fourth person goes to city B for a cost of 20. The total minimum cost is 10 + 30 + 50 + 20 = 110 to have half the people interviewing in each city.

Example 2:

Input: costs = [[259,770],[448,54],[926,667],[184,139],[840,118],[577,469]] **Output:** 1859

Example 3:

Input: costs = [[515,563],[451,713],[537,709],[343,819],[855,779],[457,60],[650,359],[631,42]] **Output:** 3086

Constraints:

* `2 * n == costs.length` * `2 <= costs.length <= 100` * `costs.length` is even. * `1 <= aCosti, bCosti <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int twoCitySchedCost(vector<vector<int>>& costs) {

}
};
```

Java:

```
class Solution {
public int twoCitySchedCost(int[][] costs) {

}
}
```

Python3:

```
class Solution:
def twoCitySchedCost(self, costs: List[List[int]]) -> int:
```