Problem 3695: Maximize Alternating Sum Using Swaps

Problem Information

Difficulty: Hard
Acceptance Rate: 63.34%
Paid Only: No
Tags: Array, Greedy, Union Find, Sorting

Problem Description

You are given an integer array `nums`.

You want to maximize the **alternating sum** of `nums`, which is defined as the value obtained by **adding** elements at even indices and **subtracting** elements at odd indices. That is, `nums[0] - nums[1] + nums[2] - nums[3]...`

You are also given a 2D integer array `swaps` where `swaps[i] = [pi, qi]`. For each pair `[pi, qi]` in `swaps`, you are allowed to swap the elements at indices `pi` and `qi`. These swaps can be performed any number of times and in any order.

Return the maximum possible **alternating sum** of `nums`.

Example 1:

Input: nums = [1,2,3], swaps = [[0,2],[1,2]]

Output: 4

Explanation:

The maximum alternating sum is achieved when `nums` is `[2, 1, 3]` or `[3, 1, 2]`. As an example, you can obtain `nums = [2, 1, 3]` as follows.

* Swap `nums[0]` and `nums[2]`. `nums` is now `[3, 2, 1]`. * Swap `nums[1]` and `nums[2]`. `nums` is now `[3, 1, 2]`. * Swap `nums[0]` and `nums[2]`. `nums` is now `[2, 1, 3]`.

Example 2:

Input: nums = [1,2,3], swaps = [[1,2]]

Output: 2

Explanation:

The maximum alternating sum is achieved by not performing any swaps.

Example 3:

Input: nums = [1,1000000000,1,1000000000,1,1000000000], swaps = []

Output: -2999999997

Explanation:

Since we cannot perform any swaps, the maximum alternating sum is achieved by not performing any swaps.

Constraints:

* `2 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `0 <= swaps.length <= 105` * `swaps[i] = [pi, qi]` * `0 <= pi < qi <= nums.length - 1` * `[pi, qi] != [pj, qj]`

Code Snippets

C++:

```
class Solution {
public:
long long maxAlternatingSum(vector<int>& nums, vector<vector<int>>& swaps) {

}
};
```

Java:

```
class Solution {
public long maxAlternatingSum(int[] nums, int[][] swaps) {

}
}
```

Python3:

```
class Solution:
def maxAlternatingSum(self, nums: List[int], swaps: List[List[int]]) -> int:
```