5505. Maximum Sum Obtained of Any Permutation

We have an array of integers, `nums` , and an array of `requests` where `requests[i]` = $[start_i, end_i]$. The ith request asks for the sum of `nums[start`$_i$`]` + `nums[start`$_i$` + 1]` + `...` + `nums[end`$_i$` − 1]` + `nums[end`$_i$`]` . Both `start`$_i$ and `end`$_i$ are *0-indexed*.

Return *the maximum total sum of all requests **among all permutations** of* `nums` .

Since the answer may be too large, return it **modulo** 10^9 `+ 7` .

User Accepted:	1
User Tried:	3
Total Accepted:	1
Total Submissions:	3
Difficulty:	Medium

Example 1:

```
Input: nums = [1,2,3,4,5], requests = [[1,3],[0,1]]
Output: 19
Explanation: One permutation of nums is [2,1,3,4,5] with the following result:
requests[0] -> nums[1] + nums[2] + nums[3] = 1 + 3 + 4 = 8
requests[1] -> nums[0] + nums[1] = 2 + 1 = 3
Total sum: 8 + 3 = 11.
A permutation with a higher total sum is [3,5,4,2,1] with the following result:
requests[0] -> nums[1] + nums[2] + nums[3] = 5 + 4 + 2 = 11
requests[1] -> nums[0] + nums[1] = 3 + 5  = 8
Total sum: 11 + 8 = 19, which is the best that you can do.
```

Example 2:

```
Input: nums = [1,2,3,4,5,6], requests = [[0,1]]
Output: 11
Explanation: A permutation with the max total sum is [6,5,4,3,2,1] with request sums [11].
```

Example 3:

```
Input: nums = [1,2,3,4,5,10], requests = [[0,2],[1,3],[1,1]]
Output: 47
Explanation: A permutation with the max total sum is [4,10,5,3,2,1] with request sums [19,
```

Constraints:

- `n == nums.length`
- `1 <= n <= 10`^{5}

- $0 <= nums[i] <= 10^5$
- $1 <= requests.length <= 10^5$
- $requests[i].length == 2$
- $0 <= start_i <= end_i < n$

JavaScript

```
1  /**
2   * @param {number[]} nums
3   * @param {number[][]} requests
4   * @return {number}
5   */
6  var maxSumRangeQuery = function(nums, requests) {
7
8  };
```

☐ **Custom Testcase**   (Use Example Testcases)

▶ Run   ☁ Submit