6424. Semi-Ordered Permutation

My Submissions (/contest/weekly-contest-348/problems/semi-ordered-permutation/submissions/) | Back to Contest (/contest/weekly-contest-348/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

You are given a **0-indexed** permutation of `n` integers `nums`.

A permutation is called **semi-ordered** if the first number equals `1` and the last number equals `n`. You can perform the below operation as many times as you want until you make `nums` a **semi-ordered** permutation:

- Pick two adjacent elements in `nums`, then swap them.

Return *the minimum number of operations to make* `nums` *a* ***semi-ordered permutation***.

A **permutation** is a sequence of integers from `1` to `n` of length `n` containing each number exactly once.

Example 1:

```
Input: nums = [2,1,4,3]
Output: 2
Explanation: We can make the permutation semi-ordered using these sequence of operations:
1 - swap i = 0 and j = 1. The permutation becomes [1,2,4,3].
2 - swap i = 2 and j = 3. The permutation becomes [1,2,3,4].
It can be proved that there is no sequence of less than two operations that make nums a semi-ordered permutation.
```

Example 2:

```
Input: nums = [2,4,1,3]
Output: 3
Explanation: We can make the permutation semi-ordered using these sequence of operations:
1 - swap i = 1 and j = 2. The permutation becomes [2,1,4,3].
2 - swap i = 0 and j = 1. The permutation becomes [1,2,4,3].
3 - swap i = 2 and j = 3. The permutation becomes [1,2,3,4].
It can be proved that there is no sequence of less than three operations that make nums a semi-ordered permutation.
```

Example 3:

```
Input: nums = [1,3,4,2,5]
Output: 0
Explanation: The permutation is already a semi-ordered permutation.
```

Constraints:

- `2 <= nums.length == n <= 50`
- `1 <= nums[i] <= 50`
- `nums is a permutation.`

JavaScript

```
const semiOrderedPermutation = (a) => {
    let n = a.length, l, r;
    for (let i = 0; i < n; i++) {
        if (a[i] == 1) {
            l = i;
        } else if (a[i] == n) {
            r = i;
        }
    }
    let disL = l, disR = n - 1 - r;
    return l < r ? disL + disR : disL + disR - 1;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/963334797/) More Details (/submissions/detail/963334797/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)