# Problem 775: Global and Local Inversions

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 42.61%
**Paid Only:** No
**Tags:** Array, Math

## Problem Description

You are given an integer array `nums` of length `n` which represents a permutation of all the integers in the range `[0, n - 1]`.

The number of **global inversions** is the number of the different pairs `(i, j)` where:

* `0 <= i < j < n` * `nums[i] > nums[j]`

The number of **local inversions** is the number of indices `i` where:

* `0 <= i < n - 1` * `nums[i] > nums[i + 1]`

Return `true` _if the number of**global inversions** is equal to the number of **local inversions**_.

**Example 1:**

**Input:** nums = [1,0,2] **Output:** true **Explanation:** There is 1 global inversion and 1 local inversion.

**Example 2:**

**Input:** nums = [1,2,0] **Output:** false **Explanation:** There are 2 global inversions and 1 local inversion.

**Constraints:**

* `n == nums.length` * `1 <= n <= 105` * `0 <= nums[i] < n` * All the integers of `nums` are **unique**. * `nums` is a permutation of all the numbers in the range `[0, n - 1]`.

## Code Snippets

**C++:**

```
class Solution {
public:
bool isIdealPermutation(vector<int>& nums) {

}
};
```

**Java:**

```
class Solution {
public boolean isIdealPermutation(int[] nums) {

}
}
```

**Python3:**

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