Problem 1053: Previous Permutation With One Swap

Problem Information

Difficulty: Medium
Acceptance Rate: 49.34%
Paid Only: No
Tags: Array, Greedy

Problem Description

Given an array of positive integers `arr` (not necessarily distinct), return _the_ _lexicographically_ _largest permutation that is smaller than_ `arr`, that can be **made with exactly one swap**. If it cannot be done, then return the same array.

Note that a _swap_ exchanges the positions of two numbers `arr[i]` and `arr[j]`

Example 1:

Input: arr = [3,2,1] **Output:** [3,1,2] **Explanation:** Swapping 2 and 1.

Example 2:

Input: arr = [1,1,5] **Output:** [1,1,5] **Explanation:** This is already the smallest permutation.

Example 3:

Input: arr = [1,9,4,6,7] **Output:** [1,7,4,6,9] **Explanation:** Swapping 9 and 7.

Constraints:

* `1 <= arr.length <= 104` * `1 <= arr[i] <= 104`

Code Snippets

C++:

```
class Solution {
public:
vector<int> prevPermOpt1(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int[] prevPermOpt1(int[] arr) {

}
}
```

Python3:

```
class Solution:
def prevPermOpt1(self, arr: List[int]) -> List[int]:
```