

Question1 : Given an integer array arr, remove a subarray (can be empty) from the array such that the remaining elements in arr are non-decreasing.

A subarray is a contiguous subsequence of the array.

Return the length of the shortest subarray to remove.

Example 1:

Input: arr = [1,2,3,10,4,2,3,5]

Output: 3

Explanation: The shortest subarray we can remove is [10,4,2] of length 3. The remaining elements after that will be [1,2,3,3,5] which are sorted.

Another correct solution is to remove the subarray [3,10,4].

Example 2:

Input: arr = [5,4,3,2,1]

Output: 4

Explanation: Since the array is strictly decreasing, we can only keep a single element. Therefore we need to remove a subarray of length 4, either [5,4,3,2] or [4,3,2,1].

Example 3:

Input: arr = [1,2,3]

Output: 0

Explanation: The array is already non-decreasing. We do not need to remove any elements.

Example 4:

Input: arr = [1]

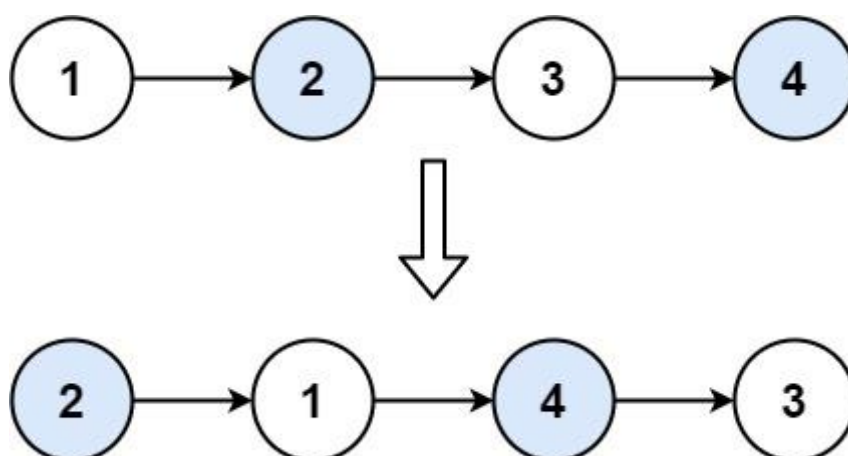
Output: 0

Constraints:

- $1 \leq \text{arr.length} \leq 10^5$
- $0 \leq \text{arr}[i] \leq 10^9$

Question 2: Given a linked list, swap every two adjacent nodes and return its head. You must solve the problem without modifying the values in the list's nodes (i.e., only nodes themselves may be changed.)

Example 1:



Input: head = [1,2,3,4]

Output: [2,1,4,3]

Example 2:

Input: head = []

Output: []

Example 3:

Input: head = [1]

Output: [1]

Constraints:

- The number of nodes in the list is in the range [0, 100].
- $0 \leq \text{Node.val} \leq 100$