969. Pancake Sorting

Given an array of integers `arr`, sort the array by performing a series of **pancake flips**.

In one pancake flip we do the following steps:

- Choose an integer `k` where `1 <= k <= arr.length`.
- Reverse the sub-array `arr[0...k-1]` (**0-indexed**).

For example, if `arr = [3,2,1,4]` and we performed a pancake flip choosing `k = 3`, we reverse the sub-array `[3,2,1]`, so `arr = [1,2,3,4]` after the pancake flip at `k = 3`.

Return *an array of the* `k`*-values corresponding to a sequence of pancake flips that sort* `arr`. Any valid answer that sorts the array within `10 * arr.length` flips will be judged as correct.

Example 1:

Input: arr = [3,2,4,1]
Output: [4,2,4,3]
Explanation:
We perform 4 pancake flips, with k values 4, 2, 4, and 3.
Starting state: arr = [3, 2, 4, 1]
After 1st flip (k = 4): arr = [1, 4, 2, 3]
After 2nd flip (k = 2): arr = [4, 1, 2, 3]
After 3rd flip (k = 4): arr = [3, 2, 1, 4]
After 4th flip (k = 3): arr = [1, 2, 3, 4], which is sorted.

Example 2:

Input: arr = [1,2,3]
Output: []
Explanation: The input is already sorted, so there is no need to flip anything.
Note that other answers, such as [3, 3], would also be accepted.

Find the largest element `A[i]`, reverse `A[0:i+1]`, making the current largest at the head of the array, then reverse the whole array to make `A[i]` at the bottom.
Do the above again and again, finally we'll have the whole array sorted.
eg:

```
[3,1,4,2] (input array)
[4,1,3,2] -> [2,3,1,4] (current maximum 4 is placed at the bottom)
[3,2,1,4] -> [1,2,3,4] (current maximum 3 is placed at the bottom)
```

```
[2,1,3,4] -> [1,2,3,4] (current maximum 2 is placed at the bottom)
[1,2,3,4] -> [1,2,3,4] (current maximum 1 is placed at the bottom)
done!
```

```python
def pancakeSort(self, A: List[int]) -> List[int]:
    n = len(A)
    res = []
    for i in range(n):
        cur_max = max(A[0:n-i])
        j = 0
        while A[j] != cur_max:
            j += 1
        # should reverse j+1 elements
        A[:j+1] = reversed(A[:j+1])
        res.append(j+1)
        # reverse all
        A[:n-i] = reversed(A[:n-i])
        res.append(n-i)
    return res
```