Algorithm Question - In-Place Maximum Marking

Problem Statement:

You are given a list of integers " arr " where arr = [4, 9, 9, 8, 3]. Your task is to write a Python function that modifies " arr ".

The output should be **arr = [9, 4, 1, 9, 3]** , where only the first max moves to index 0.

Your algorithm should respect to these conditions:

1. The function must not return anything — the caller should see the change because of Python's object reference behaviour.
2. You are not allowed to create a new list and array.
3. The algorithm must be efficient and must correctly handle lists with duplicate maximum values.
4. The algorithm must work even if the list contains negative numbers.
5. The algorithm must run in O(n) or O(log n) time.

Notes:

1. O(n) → Linear Time (time grows in direct proportion to input size).
2. O(log n) → logarithmic time (super-efficient, like binary search).\

Hints:

1. Reassigning the parameter (e.g. " arr = something ") will not affect the original list, but changing elements inside it (e.g. " arr[0] = x ") will.
2. Start by assuming the first element is the largest.
3. If you find something bigger, swap both the largest value and its index.