Problem 1:

You are given a string `s`, which contains stars `*`.

In one operation, you can:

- Choose a star in `s`.
- Remove the closest **non-star** character to its **left**, as well as remove the star itself.

Return *the string after **all** stars have been removed.*

Note:

- The input will be generated such that the operation is always possible.
- It can be shown that the resulting string will always be unique.

Example 1:

```
Input: s = "leet**cod*e"
Output: "lecoe"
Explanation: Performing the removals from left to right:
- The closest character to the 1st star is 't' in "leet**cod*e". s becomes "lee*cod*e".
- The closest character to the 2nd star is 'e' in "lee*cod*e". s becomes "lecod*e".
- The closest character to the 3rd star is 'd' in "lecod*e". s becomes "lecoe".
There are no more stars, so we return "lecoe".
```

Example 2:

```
Input: s = "erase*****"
Output: ""
Explanation: The entire string is removed, so we return an empty string.
```

Problem 2:

You can choose two integers `left` and `right` where `0 <= left <= right < n` and **swap** the subarray `nums1[left...right]` with the subarray `nums2[left...right]`.

- For example, if `nums1 = [1,2,3,4,5]` and `nums2 = [11,12,13,14,15]` and you choose `left = 1` and `right = 2`, `nums1` becomes `[1,12,13,4,5]` and `nums2` becomes `[11,2,3,14,15]`.

You may choose to apply the mentioned operation **once** or not do anything.

The **score** of the arrays is the **maximum** of `sum(nums1)` and `sum(nums2)`, where `sum(arr)` is the sum of all the elements in the array `arr`.

Return *the **maximum possible score***.

A **subarray** is a contiguous sequence of elements within an array. `arr[left...right]` denotes the subarray that contains the elements of `nums` between indices `left` and `right` (**inclusive**).

Example 1:

```
Input: nums1 = [60,60,60], nums2 = [10,90,10]
Output: 210
Explanation: Choosing left = 1 and right = 1, we have nums1 = [60,90,60] and nums2 = [10,60,10].
The score is max(sum(nums1), sum(nums2)) = max(210, 80) = 210.
```

Example 2:

```
Input: nums1 = [20,40,20,70,30], nums2 = [50,20,50,40,20]
Output: 220
Explanation: Choosing left = 3, right = 4, we have nums1 = [20,40,20,40,20] and nums2 =
[50,20,50,70,30].
The score is max(sum(nums1), sum(nums2)) = max(140, 220) = 220.
```

Example 3:

```
Input: nums1 = [7,11,13], nums2 = [1,1,1]
Output: 31
Explanation: We choose not to swap any subarray.
The score is max(sum(nums1), sum(nums2)) = max(31, 3) = 31.
```