Problem 2089: Find Target Indices After Sorting Array

Problem Information

Difficulty: Easy
Acceptance Rate: 77.63%
Paid Only: No
Tags: Array, Binary Search, Sorting

Problem Description

You are given a **0-indexed** integer array `nums` and a target element `target`.

A **target index** is an index `i` such that `nums[i] == target`.

Return _a list of the target indices of_ `nums` after _sorting_ `nums` _in**non-decreasing** order_. If there are no target indices, return _an**empty** list_. The returned list must be sorted in **increasing** order.

Example 1:

Input: nums = [1,2,5,2,3], target = 2 **Output:** [1,2] **Explanation:** After sorting, nums is [1,_**2**_ ,_**2**_ ,3,5]. The indices where nums[i] == 2 are 1 and 2.

Example 2:

Input: nums = [1,2,5,2,3], target = 3 **Output:** [3] **Explanation:** After sorting, nums is [1,2,2,_**3**_ ,5]. The index where nums[i] == 3 is 3.

Example 3:

Input: nums = [1,2,5,2,3], target = 5 **Output:** [4] **Explanation:** After sorting, nums is [1,2,2,3,_**5**_]. The index where nums[i] == 5 is 4.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i], target <= 100`

Code Snippets

C++:

```
class Solution {
public:
vector<int> targetIndices(vector<int>& nums, int target) {

}
};
```

Java:

```
class Solution {
public List<Integer> targetIndices(int[] nums, int target) {

}
}
```

Python3:

```
class Solution:
def targetIndices(self, nums: List[int], target: int) -> List[int]:
```