Problem 1673: Find the Most Competitive Subsequence

Problem Information

Difficulty: Medium
Acceptance Rate: 52.39%
Paid Only: No
Tags: Array, Stack, Greedy, Monotonic Stack

Problem Description

Given an integer array `nums` and a positive integer `k`, return _the most**competitive** subsequence of _`nums` _of size_`k`.

An array's subsequence is a resulting sequence obtained by erasing some (possibly zero) elements from the array.

We define that a subsequence `a` is more **competitive** than a subsequence `b` (of the same length) if in the first position where `a` and `b` differ, subsequence `a` has a number **less** than the corresponding number in `b`. For example, `[1,3,4]` is more competitive than `[1,3,5]` because the first position they differ is at the final number, and `4` is less than `5`.

Example 1:

Input: nums = [3,5,2,6], k = 2 **Output:** [2,6] **Explanation:** Among the set of every possible subsequence: {[3,5], [3,2], [3,6], [5,2], [5,6], [2,6]}, [2,6] is the most competitive.

Example 2:

Input: nums = [2,4,3,3,5,4,9,6], k = 4 **Output:** [2,3,3,4]

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `1 <= k <= nums.length`

Code Snippets

C++:

```
class Solution {
public:
vector<int> mostCompetitive(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int[] mostCompetitive(int[] nums, int k) {

}
}
```

Python3:

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