Problem 2099: Find Subsequence of Length K With the Largest Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 57.44%
Paid Only: No
Tags: Array, Hash Table, Sorting, Heap (Priority Queue)

Problem Description

You are given an integer array `nums` and an integer `k`. You want to find a **subsequence** of `nums` of length `k` that has the **largest** sum.

Return ___**any** such subsequence as an integer array of length _`k`.

A **subsequence** is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

Example 1:

Input: nums = [2,1,3,3], k = 2 **Output:** [3,3] **Explanation:** The subsequence has the largest sum of 3 + 3 = 6.

Example 2:

Input: nums = [-1,-2,3,4], k = 3 **Output:** [-1,3,4] **Explanation:** The subsequence has the largest sum of -1 + 3 + 4 = 6.

Example 3:

Input: nums = [3,4,3,3], k = 2 **Output:** [3,4] **Explanation:** The subsequence has the largest sum of 3 + 4 = 7. Another possible subsequence is [4, 3].

Constraints:

* `1 <= nums.length <= 1000` * `-105 <= nums[i] <= 105` * `1 <= k <= nums.length`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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