Problem 2386: Find the K-Sum of an Array

Problem Information

Difficulty: Hard
Acceptance Rate: 40.53%
Paid Only: No
Tags: Array, Sorting, Heap (Priority Queue)

Problem Description

You are given an integer array `nums` and a **positive** integer `k`. You can choose any **subsequence** of the array and sum all of its elements together.

We define the **K-Sum** of the array as the `kth` **largest** subsequence sum that can be obtained (**not** necessarily distinct).

Return _the K-Sum of the array_.

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.

Note that the empty subsequence is considered to have a sum of `0`.

Example 1:

Input: nums = [2,4,-2], k = 5 **Output:** 2 **Explanation:** All the possible subsequence sums that we can obtain are the following sorted in decreasing order: 6, 4, 4, 2, _2_ , 0, 0, -2. The 5-Sum of the array is 2.

Example 2:

Input: nums = [1,-2,3,4,-10,12], k = 16 **Output:** 10 **Explanation:** The 16-Sum of the array is 10.

Constraints:

* `n == nums.length` * `1 <= n <= 105` * `-109 <= nums[i] <= 109` * `1 <= k <= min(2000, 2n)`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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