Problem 373: Find K Pairs with Smallest Sums

Problem Information

Difficulty: Medium
Acceptance Rate: 41.16%
Paid Only: No
Tags: Array, Heap (Priority Queue)

Problem Description

You are given two integer arrays `nums1` and `nums2` sorted in **non- decreasing order** and an integer `k`.

Define a pair `(u, v)` which consists of one element from the first array and one element from the second array.

Return _the_ `k` _pairs_ `(u1, v1), (u2, v2), ..., (uk, vk)` _with the smallest sums_.

Example 1:

Input: nums1 = [1,7,11], nums2 = [2,4,6], k = 3 **Output:** [[1,2],[1,4],[1,6]]
Explanation: The first 3 pairs are returned from the sequence:
[1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]

Example 2:

Input: nums1 = [1,1,2], nums2 = [1,2,3], k = 2 **Output:** [[1,1],[1,1]] **Explanation:** The first 2 pairs are returned from the sequence: [1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]

Constraints:

* `1 <= nums1.length, nums2.length <= 105` * `-109 <= nums1[i], nums2[i] <= 109` * `nums1` and `nums2` both are sorted in **non-decreasing order**. * `1 <= k <= 104` * `k <= nums1.length * nums2.length`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> kSmallestPairs(vector<int>& nums1, vector<int>& nums2,
int k) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> kSmallestPairs(int[] nums1, int[] nums2, int k) {

}
}
```

Python3:

```
class Solution:
def kSmallestPairs(self, nums1: List[int], nums2: List[int], k: int) ->
List[List[int]]:
```