Problem 632: Smallest Range Covering Elements from K Lists

Problem Information

Difficulty: Hard
Acceptance Rate: 69.93%
Paid Only: No
Tags: Array, Hash Table, Greedy, Sliding Window, Sorting, Heap (Priority Queue)

Problem Description

You have `k` lists of sorted integers in **non-decreasing order**. Find the **smallest** range that includes at least one number from each of the `k` lists.

We define the range `[a, b]` is smaller than range `[c, d]` if `b - a < d - c` **or** `a < c` if `b - a == d - c`.

Example 1:

Input: nums = [[4,10,15,24,26],[0,9,12,20],[5,18,22,30]] **Output:** [20,24] **Explanation:** List 1: [4, 10, 15, 24,26], 24 is in range [20,24]. List 2: [0, 9, 12, 20], 20 is in range [20,24]. List 3: [5, 18, 22, 30], 22 is in range [20,24].

Example 2:

Input: nums = [[1,2,3],[1,2,3],[1,2,3]] **Output:** [1,1]

Constraints:

* `nums.length == k` * `1 <= k <= 3500` * `1 <= nums[i].length <= 50` * `-105 <= nums[i][j] <= 105` * `nums[i]` is sorted in **non-decreasing** order.

Code Snippets

C++:

```
class Solution {
public:
vector<int> smallestRange(vector<vector<int>>& nums) {

}
};
```

Java:

```
class Solution {
public int[] smallestRange(List<List<Integer>> nums) {

}
}
```

Python3:

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