632. Smallest Range Covering Elements from K Lists

You have `k` lists of sorted integers in ascending 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: [[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].
```

Note:

1. The given list may contain duplicates, so ascending order means >= here.
2. 1 <= `k` <= 3500
3. -10^5 <= `value of elements` <= 10^5.

CODE:

```
vector<int> smallestRange(vector<vector<int>>& nums) {
    int rows = nums.size();
priority_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>> pq;

      int maxe = -1;

    for(int i=0;i<rows;i++){
       pq.push({nums[i][0],i});
       maxe = max(maxe,nums[i][0]);
      }

      int minRange = INT_MAX;
      vector<int> minInd(rows,0);
      vector<int> ans;

    while(true){
      pair<int,int> minp = pq.top();    pq.pop();

      if(maxe - minp.first < minRange){
         minRange = maxe - minp.first;
         ans = {minp.first,maxe};
      }

      minInd[minp.second]++;

      if(minInd[minp.second] >= nums[minp.second].size())
         break;
```

```
    maxe = max(maxe,nums[minp.second][minInd[minp.second]]);
    pq.push({nums[minp.second][minInd[minp.second]],minp.second});
  }
return ans;
}
```