Problem 1439: Find the Kth Smallest Sum of a Matrix With Sorted Rows

Problem Information

Difficulty: Hard
Acceptance Rate: 62.18%
Paid Only: No
Tags: Array, Binary Search, Heap (Priority Queue), Matrix

Problem Description

You are given an `m x n` matrix `mat` that has its rows sorted in non- decreasing order and an integer `k`.

You are allowed to choose **exactly one element** from each row to form an array.

Return _the_`kth` _smallest array sum among all possible arrays_.

Example 1:

Input: mat = [[1,3,11],[2,4,6]], k = 5 **Output:** 7 **Explanation:** Choosing one element from each row, the first k smallest sum are: [1,2], [1,4], [3,2], [3,4], [1,6]. Where the 5th sum is 7.

Example 2:

Input: mat = [[1,3,11],[2,4,6]], k = 9 **Output:** 17

Example 3:

Input: mat = [[1,10,10],[1,4,5],[2,3,6]], k = 7 **Output:** 9 **Explanation:** Choosing one element from each row, the first k smallest sum are: [1,1,2], [1,1,3], [1,4,2], [1,4,3], [1,1,6], [1,5,2], [1,5,3]. Where the 7th sum is 9.

Constraints:

* `m == mat.length` * `n == mat.length[i]` * `1 <= m, n <= 40` * `1 <= mat[i][j] <= 5000` * `1 <= k <= min(200, nm)` * `mat[i]` is a non-decreasing array.

Code Snippets

C++:

```
class Solution {
public:
int kthSmallest(vector<vector<int>>& mat, int k) {

}
};
```

Java:

```
class Solution {
public int kthSmallest(int[][] mat, int k) {

}
}
```

Python3:

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