Problem 2392: Build a Matrix With Conditions

Problem Information

Difficulty: Hard
Acceptance Rate: 79.30%
Paid Only: No
Tags: Array, Graph, Topological Sort, Matrix

Problem Description

You are given a **positive** integer `k`. You are also given:

* a 2D integer array `rowConditions` of size `n` where `rowConditions[i] = [abovei, belowi]`, and * a 2D integer array `colConditions` of size `m` where `colConditions[i] = [lefti, righti]`.

The two arrays contain integers from `1` to `k`.

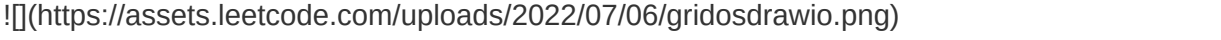
You have to build a `k x k` matrix that contains each of the numbers from `1` to `k` **exactly once**. The remaining cells should have the value `0`.

The matrix should also satisfy the following conditions:

* The number `abovei` should appear in a **row** that is strictly **above** the row at which the number `belowi` appears for all `i` from `0` to `n - 1`. * The number `lefti` should appear in a **column** that is strictly **left** of the column at which the number `righti` appears for all `i` from `0` to `m - 1`.

Return _**any** matrix that satisfies the conditions_. If no answer exists, return an empty matrix.

Example 1:



Input: k = 3, rowConditions = [[1,2],[3,2]], colConditions = [[2,1],[3,2]] **Output:** [[3,0,0],[0,0,1],[0,2,0]] **Explanation:** The diagram above shows a valid example of a matrix

that satisfies all the conditions. The row conditions are the following: - Number 1 is in row _1_ , and number 2 is in row _2_ , so 1 is above 2 in the matrix. - Number 3 is in row _0_ , and number 2 is in row _2_ , so 3 is above 2 in the matrix. The column conditions are the following: - Number 2 is in column _1_ , and number 1 is in column _2_ , so 2 is left of 1 in the matrix. - Number 3 is in column _0_ , and number 2 is in column _1_ , so 3 is left of 2 in the matrix. Note that there may be multiple correct answers.

Example 2:

Input: k = 3, rowConditions = [[1,2],[2,3],[3,1],[2,3]], colConditions = [[2,1]] **Output:** [] **Explanation:** From the first two conditions, 3 has to be below 1 but the third conditions needs 3 to be above 1 to be satisfied. No matrix can satisfy all the conditions, so we return the empty matrix.

Constraints:

* `2 <= k <= 400` * `1 <= rowConditions.length, colConditions.length <= 104` * `rowConditions[i].length == colConditions[i].length == 2` * `1 <= abovei, belowi, lefti, righti <= k` * `abovei != belowi` * `lefti != righti`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> buildMatrix(int k, vector<vector<int>>& rowConditions,
vector<vector<int>>& colConditions) {

}
};
```

Java:

```
class Solution {
public int[][] buildMatrix(int k, int[][] rowConditions, int[][]
colConditions) {

}
}
```

Python3:

```
class Solution:
def buildMatrix(self, k: int, rowConditions: List[List[int]], colConditions:
List[List[int]]) -> List[List[int]]:
```