Difficulty:

(Hard)

(1) Evalera (Javalera) Drahlama (Javahlama et all) Interview Contact Discuss (Jaine September 1) 10 A 110 Profile Prof	(/) Explore(/explore/)	Problems(/problemset/aii/)	interview	Contest	Store (https://leetcode.com/jobs/) (/subscribe?	-	diagonally/
	(/) Explore(/explore/)	Problems(/problemset/all/)	Interview	Contest	Discuss(/discuss/) eet de in hiring land (subscribe?	ţ	∘ othe-matrix

6163. Build a Matrix With Conditions

My Submissions (/contest/weekly-contest-308/problems/build-a-matrix-with-conditions/submissions/) Back to Contest (/contest/weekly-contest-308/) You are given a **positive** integer k. You are also given: User Accepted: 0 • a 2D integer array rowConditions of size n where rowConditions[i] = $[above_i, below_i]$, and User Tried: 3 • a 2D integer array colConditions of size m where colConditions[i] = [left_i, right_i]. 0 The two arrays contain integers from 1 to k. Total Accepted: 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 **Total Submissions:** 3 have the value 0.

The matrix should also satisfy the following conditions:

- The number above i should appear in a row that is strictly above the row at which the number below appears for all i from 0 to n 1.
- The number left; should appear in a column that is strictly left of the column at which the number right; 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:

3	0	0
0	0	1
0	2	0

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 <= above $_i$, below $_i$, left $_i$, right $_i$ <= k
- above_i != below_i
- left $_i$!= right $_i$



1 const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };

```
2 | const packDGInDegree = (g, edges, indegree) => { for (const [u, v] of edges) { g[u-1].unshift(v-1); indegree[v-1]++; } };
    const initialize2DArray = (n, m) \Rightarrow \{ let d = []; for (let i = 0; i < n; i++) \{ let t = Array(m).fill(0); d.push(t); \} \}
    return d; };
 5 ,
    const buildMatrix = (k, rowConditions, colConditions) => {
6
        let gr = make(k, rowConditions), gc = make(k, colConditions), d = initialize2DArray(k, 2), res = initialize2DArray(k, 2)
    k);
7
        if (gr.length == 0 || gc.length == 0) return [];
8
        for (let i = 0; i < k; i++) {
            d[gr[i]][0] = i;
9
10
            d[gc[i]][1] = i;
11
        for (let i = 0; i < k; i++) {
12 1
13
            let [x, y] = d[i];
14
             res[x][y] = i + 1;
15
16
        return res;
17
    };
18
19 ▼
    const make = (n, edges) \Rightarrow {
20
        let g = initializeGraph(n), deg = Array(n).fill(0);
21
        packDGInDegree(g, edges, deg);
22
        return topologicalSort(g, deg);
23
    };
24
25 ▼
    const topologicalSort = (g, indegree) => {
26
        let res = [], q = [], n = g.length;
        for (let i = 0; i < n; i++) {
27
             if (indegree[i] == 0) q.push(i);
28
29
30 ▼
        while (q.length) {
31
            let cur = q.shift();
32
             res.push(cur);
             for (const child of g[cur]) {
33 •
34
                 indegree[child]--;
35
                 if (indegree[child] == 0) q.push(child);
36
37
38 ▼
        for (let i = 0; i < n; i++) {
39
             if (indegree[i] > 0) return [];
40
41
        return res;
    };
42
```

☐ Custom Testcase

Use Example Testcases

Run

Submit
 Su

Submission Result: Accepted (/submissions/detail/785291734/) ?

More Details > (/submissions/detail/785291734/)

Share your acceptance!

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)