

Getting started with Competitive Programming

Week – 9 Network flow: min cut

Problem: Covering chessboard

Link - <https://www.hackerearth.com/practice/algorithms/graphs/min-cut/practice-problems/algorithm/covering-chessboard/>

You have an n by m grid. Each grid square has a certain value associated with it. This is given by the numbers $v_{i,j}$.

You can capture grid squares in two different ways.

1. You can *directly* capture a grid square. This costs $c_{i,j}$.
2. You can *indirectly* capture a grid square. You can only do this if we have already captured all of this squares neighbors. This costs $b_{i,j} \leq c_{i,j}$.

Two squares are neighbors if they share an edge.

Your score is the sum of values you have captured minus the cost it took to capture those squares. Return the maximum possible score you can achieve.

Input format:

The first line of input will contain two integers n, m .

The next n lines of input will contain m space separated integers each. The j -th number on the i -th line denotes $v_{i,j}$.

The next n lines of input will contain m space separated integers each. The j -th number on the i -th line denotes $b_{i,j}$.

The next n lines of input will contain m space separated integers each. The j -th number on the i -th line denotes $c_{i,j}$.

Output format:

Print a single integer, the answer to the problem.

Constraints

For all subtasks:

$$0 \leq v_{i,j} \leq 100,000$$

$$0 \leq b_{i,j} \leq c_{i,j} \leq 100,000$$

$$1 \leq n, m$$

Subtask 1 (45 pts):

$$n, m \leq 3$$

Subtask 2 (35 pts):

$n, m \leq 7$

Subtask 3 (20 pts):

$n, m \leq 25$

Sample Input

```
3 5
9 0 3 2 4
0 2 8 6 2
5 3 4 1 3
5 1 7 9 5
2 4 1 2 4
9 1 2 2 0
9 1 8 9 7
2 7 3 2 9
9 1 9 8 2
```

Sample Output

13

Time Limit: 2

Memory Limit: 256

Source Limit:

Explanation

A more human-readable version of the input is as follows:

```
3 5
9 0 3 2 4
0 2 8 6 2
5 3 4 1 3

5 1 7 9 5
2 4 1 2 4
9 1 2 2 0
9 1 8 9 7
2 7 3 2 9
9 1 9 8 2
```

The optimal strategy in this case is to directly capture the following squares:

```
0 X 0 0 0
X 0 X X 0
0 X 0 0 X
```

And indirectly capture only the top left square.