3 Advanced Problem: Stock Charts

We strongly recommend you start solving advanced problems only when you are done with the basic problems (for some advanced problems, algorithms are not covered in the video lectures and require additional ideas to be solved; for some other advanced problems, algorithms are covered in the lectures, but implementing them is a more challenging task than for other problems).

Problem Introduction

We would like to thank the organizers of Google Code Jam — the annual worldwide programming competition held by Google — for allowing us to use this problem from one of the past rounds. Yes, that's right: you have a problem from a world championship in programming in your first homework, but hey, this is the fifth course in the Data Structures and Algorithms Specialization, and it has the words "Advanced Algorithms" in its name for a reason :) You've made it thus far — you can do this.

In this problem you will need to guess how to apply the algorithms from this module to find the most compact way of visualizing stock price data using charts.

Problem Description

Task. You're in the middle of writing your newspaper's end-of-year economics summary, and you've decided that you want to show a number of charts to demonstrate how different stocks have performed over the course of the last year. You've already decided that you want to show the price of n different stocks, all at the same k points of the year.

A simple chart of one stock's price would draw lines between the points $(0, price_0), (1, price_1), \ldots, (k-1, price_{k-1})$, where $price_i$ is the price of the stock at the i-th point in time.

In order to save space, you have invented the concept of an overlaid chart. An overlaid chart is the combination of one or more simple charts, and shows the prices of multiple stocks (simply drawing a line for each one). In order to avoid confusion between the stocks shown in a chart, **the lines in an overlaid chart may not cross or touch**.

Given a list of n stocks' prices at each of k time points, determine the minimum number of overlaid charts you need to show all of the stocks' prices.

Input Format. The first line of the input contains two integers n and k — the number of stocks and the number of points in the year which are common for all of them. Each of the next n lines contains k integers. The i-th of those n lines contains the prices of the i-th stock at the corresponding k points in the year.

Constraints. $1 \le n \le 100$; $1 \le k \le 25$. All the stock prices are between 0 and 1 000 000.

Output Format. Output a single integer — the minimum number of overlaid charts to visualize all the stock price data you have.

Time Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time in seconds	2	2	3	10	3	4	10	10	6

Memory Limit. 512Mb.

Sample 1.

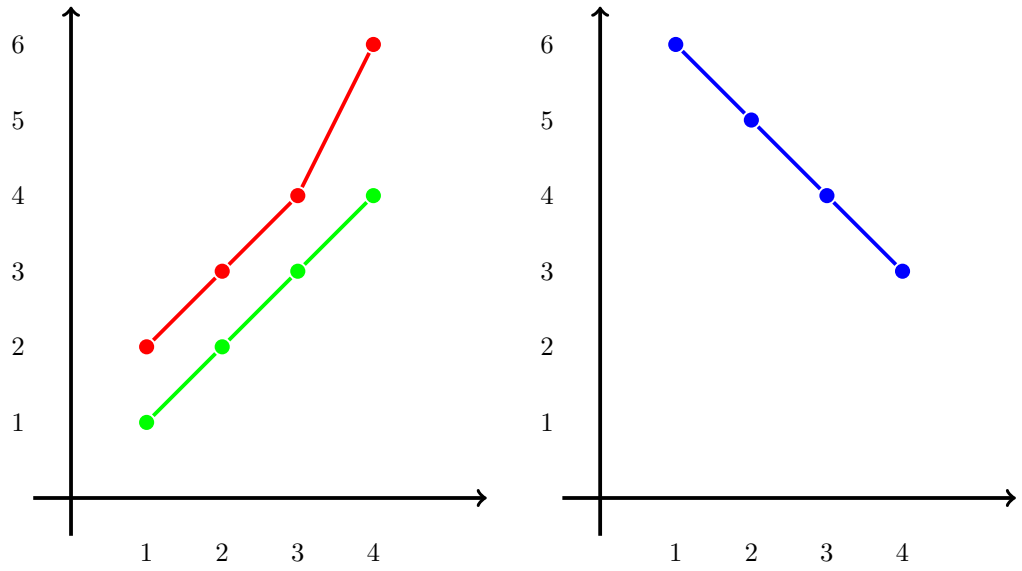
Input:

```
3 4
1 2 3 4
2 3 4 6
6 5 4 3
```

Output:

```
2
```

Explanation:
This data can be put into two following overlaid charts:



However, we cannot put all the data in one overlaid chart, as the lines corresponding to the third stock would touch the lines corresponding to the second stock, because they have the same price value at the third point.

Sample 2.

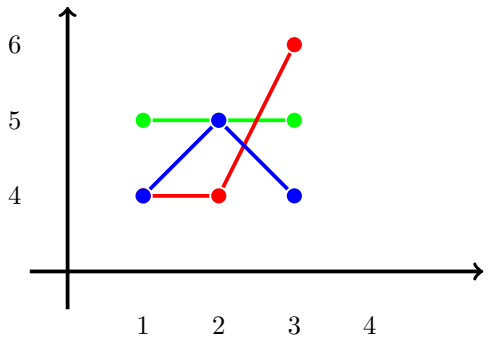
Input:

```
3 3
5 5 5
4 4 6
4 5 4
```

Output:

```
3
```

Explanation:
Each stock can be put on its own overlaid stock chart, of course. But no two stocks can be put on the same overlaid stock chart: first and second would intersect between points 2 and 3, first and third would touch in point 2, second and third would touch in point 1.



Starter Files

The starter solutions for this problem read the data from the input, pass it to a procedure and output the result. This procedure tries to pack the stock price data into the minimum possible number of overlaid charts using a greedy algorithm, but it sometimes fails. You need to implement another algorithm in this procedure if you are using `C++`, `Java`, or `Python3`. For other programming languages, you need to implement a solution from scratch. Filename: `stock_charts`

What To Do

Determine what are the conditions under which two stocks can be put on the same chart. Then think when more than 2 stocks can be put on the same chart.

Try to reduce the problem to the maximum matching in a bipartite graph problem, and to the first trick on the way is to create a bipartite graph of stocks with **two nodes for each stock**.

Need Help?

Ask a question or see the questions asked by other learners at this forum thread.