Google Code Jam Practice Problems

A - Big City Skyline

You've just moved to the Big City so you could start work at the newest Google office, Google BC, and the one thing that interests you most is the beautiful skyline. You're sitting on a hillside ten miles away, looking at the big rectangular buildings that make up the city, and you think back to your childhood.

When you were a child, you used to make toy cities out of rectangular blocks. Each building could be made from multiple blocks, and each block could be part of multiple buildings. Looking at the skyline, you wonder: what is the biggest possible block that could be part of the Big City's skyline?

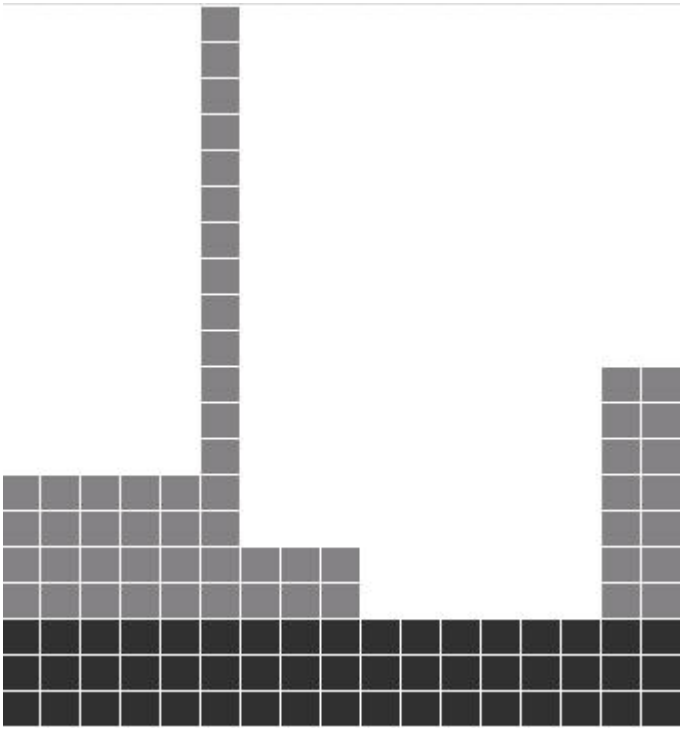
Write a program that takes the description of the skyline as an input, and gives the area of the maximum-area rectangle as output. This program should take less than 4 minutes to run on a 2GHz computer with 512MB of RAM, even for the biggest input size we specify.

Input:

The city is made out of rectangular buildings, all next to each other. The input will consist of an integer **N**, the number of buildings, followed by a series of N number pairs $\mathbf{w_i}$ $\mathbf{h_i}$, indicating the width and height of each building in order from left to right. The buildings' heights will be less than 100,000,000, and their widths will be less than 1000.

Easy: 0 < N < 1,000
Hard: 0 < N < 10,000,000

Output:
A single number: the area of the largest possible block. (Here's the beautiful skyline of the Big City.)



Sample Input	Sample Output	Explanation
5 5 7 1 20 3 5 6 3 2 10	51	The biggest block is shaded black in the figure above.

B - Universal Search

You've just started your job at Google Moon, and you've been assigned to the universal search project: that is, of course, the functionality that lets people search for anything in the universe. Your starter project seems simple: find a flight path for the GCrawler (Galaxy Crawler) that will gather stock prices from all of the galactic stock exchanges, traveling between planets using Hyperspace Jumpgates.

The known universe contains **k** stock exchanges and **p** inhabited planets. Each planet deals with one stock exchange, and every planet has a record of the stock prices for its exchange. Given a list of stock exchanges, their planets, and planet<->planet jumpgate connections, find a path for the GCrawler that contains the smallest number of jumpgates, but still passes through at least one planet from each stock exchange before returning home.

Input:

The input will start with **p**, the number of planets. Next will follow **p** lines of the form:
planet_name stock_exchange connection1 connection2 ... connectionN

If planet A connects to planet B, then planet B also connects to planet A. Planet and stock exchange names will contain only alphabetic characters [A-Za-z]. One of the planets will be named GoogleMoon; that is the start and end location for the GCrawler.
Easy: 0 < p < 50, 0 < k < 5.
Hard: 0 < p < 500, 0 < k < 16.

Output:

A series of planet names, separated by spaces, describing the path the GCrawler should take.

Sample Input
5 GoogleMoon TheInternet Mars Earth Mars EarthStocks GoogleMoon AlphaCentauriII AlphaCetiV Earth EarthStocks GoogleMoon AlphaCentauriII AlphaStocks Mars AlphaCetiV AlphaCetiV AlphaStocks Mars AlphaCentauriII
Sample Output
GoogleMoon Mars AlphaCetiV Mars GoogleMoon
Explanation
The GCrawler has three stock exchanges to visit: TheInternet, EarthStocks and AlphaStocks. It starts on GoogleMoon, which deals with TheInternet; it then travels to Mars, which trades in EarthStocks, then to AlphaCetiV, which deals in AlphaStocks. Now it needs to get back home; the shortest route goes through Mars.