South African
Computer Olympiad Training
DAY 2

Overview

Author	**Marco**	**Linsen**	**Nick**	**Richard**
Problem	**Network**	**Tunnels**	**Mafia**	**Lights**
Executable	`network.exe`	`tunnels.exe`	`mafia.exe`	`lights.exe`
Source	`network.pas` `network.cpp`	`tunnels.pas` `tunnels.cpp`	`mafia.pas` `mafia.cpp`	`lights.pas` `lights.cpp`
Input file	`network.in`	`tunnels.in`	`mafia.in`	`lights.in`
Output files(10)	`network.out`	`tunnels.out`	`mafia.out`	`lights.out`
Time limit	1 second	1 seconds	2 seconds	1 second
Num. of tests	10	10	10	10
Points per test	10	10	10	10
Total points	**100**	**100**	**100**	**100**

The maximum total score for Day2 is 300 points.

Network

Author

Marco Gallotta

Introduction

The Guji tribe has just discovered computers. They are so fascinated by them that want to make their own network of computers, with each hut having its own computer. Having limited money, they want to connect all the computers together in the cheapest way possible, using the least amount of optical fibre possible.

The Guji tribe is a strange tribe though, and they are making their lives difficult. They have said that instead of connecting every computer to the network, they want to leave one computer out of the network. More than this, they want the computer to be selected in such a way that the amount of optical fibre used is again minimised.

Task

Given a list of how much fiber it takes to connect each pair of computers, you must find the minimum amount of fiber needed to connect all but one together. This minimum network is simply the minimum of all possible networks with exactly one computer left out. Each computer in the network must connect to all other computers in the network such that a packet can flow from any one computer to another. The triangle inequality may not hold (AB + BC may be less than AC), since the Guji tribe isn't the cleverest of tribes.

Example

Assume there are 4 computers with the following length of fiber required to link them:

	0	1	2	3
0	-	4	9	21
1	4	-	8	17
2	9	8	-	16
3	21	17	16	-

With computer 0, 1, 2, and 3 excluded from the network, the minimum amounts of fiber required are 24, 25, 21 and 12 respectively. Therefore the minimum amount of fiber required is 12, when computer 3 is excluded from the network. This is achieved by connecting computers 0 and 1 (4 units) and computers 1 and 2 (8 units).

Input (network.in)

The first line contains a single integer N, the number of huts. In the next N^2 lines the fiber lengths are listed in the order they appear in the table above, going left to right then down, as in English.

Sample Input:

```
4
0
4
9
21
4
0
8
17
9
8
0
16
21
17
16
0
```

Output (network.out)

The first line of the input file should be a number from 0 to N - 1, the number of the computer removed from the network. If there is more than one computer that gives equal minimum fiber length, then the computer that appears first in the input file of these computers must be output. The second line of the output file should contain a single integer, the length that is the sum of the minimum length of fiber required to connect the entire set of computers except for one.

Sample output:

```
3
12
```

Constraints

- $3 \leq N \leq 100$
- The sum of all lengths in the input file can fit into a 32 bit signed int.

Time limit

1 second.

Scoring

- You will score 30% for only choosing the correct computer to remove from the network.
- You will score 100% if you choose the correct computer to be removed and you output the minimum length of fiber required.
- Otherwise you will score 0%.

Tunnels

Author

Linsen Loots (inspired by CEOI 2000)

Introduction

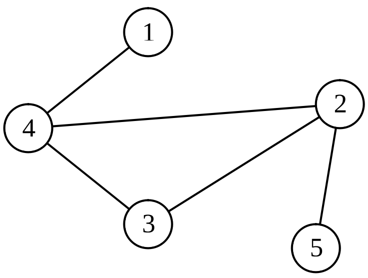
The friendly ant colony has acquired numerous enemies over the years, and these enemies have begun sabotaging their tunnels. Fortunately the ants are very well organised, and can repair any damaged tunnel very quickly. Their enemies are also not very well coordinated, so no more than one tunnel will be damaged at any one time. Unfortunately, it is essential that the ants can reach a number of locations at any given time. Being wise ants with considerable foresight, they have decided to construct additional backup tunnels to ensure that traffic flow cannot be stopped.

Task

Given the current nature of the tunnel layout, you must find out the minimum number of tunnels that need to be dug, and between which locations they will be needed. After these tunnels have been constructed, all the locations must remain connected after any one tunnel is removed. All locations will be connected in the input data, but won't all have backup routes. All tunnels function in both directions.

Example

There are 5 locations essential to the ants' survival, with the following layout:



In this example, only one tunnel needs to be constructed, namely one between location 1 and location 5. If this tunnel is built, there is no tunnel that can be removed to disconnect the locations from each other.

Input (tunnels.in)

The first line of the input will contain two space-separated integers, L and T, the number of locations and tunnels respectively.

The next T lines will each contain two space-separated integers, representing two locations that are currently connected by a tunnel. The locations are numbered from 1 to L. Two locations can only have one tunnel directly between them, and no tunnel connects a location to itself.

Sample Input:

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

Output (tunnels.out)

The first line of the output file should contain one integer, N, the number of backup tunnels to be dug.

The next N lines should each contain two space-separated integers representing two locations that need to be connected by a tunnel. You may build tunnels that connect the same pair of points as some existing tunnel, but you may not build a tunnel that connects a point to itself.

Sample output:

```
1
1 5
```

Constraints

3 <= L <= 100
T <= 1000

Time limit

1 second.

Scoring

If the given output is incorrectly formatted, or does not give a situation in which all locations remain connected when a tunnel is removed, it scores 0.

If the output is valid but not optimal, and the optimal number of tunnels is O, the score is (O/N) * 8, rounded down.

If the output is both valid and optimal, it scores 10.

Mafia

Author

Nick Pilkington

Introduction

You have become involved with various gangs including the mafia, triads and KGB through asking them for various favors. Now it's their turn! Each of the gangs are requesting protection money from you. This money is to ensure that you don't suffer from any unfortunate accidents in the near future. Being business minded each gang has given you a number of offers. Each offer consists of a price and a protection period. The price is the cost of the offer which you will have to pay, and the protection period is the amount of time for which you are assured no accidents will happen. You must purchase one (no more, no less) offer from each gang. Any excess money that you have available after your purchases is tragically stolen by a pickpocket. Any protection gained through an offer is from all the gangs so it is cumulative. You have sold all your earthly possessions in order to gain the maximum amount of money you can. Now your goal is simple: calculate the maximum amount of time that you can guarantee your own survival!

Task

You are told the number of gangs, the amount of money you can spend and descriptions of the price and period of each offer available from each gang. You MUST make ONE purchase from EACH gang! It will always be possible to make a purchase from each gang.

Example

Consider the following proposals from two gangs.

	Gang #1		Gang #2	
1	R1	4	R4	2
2			R3	3

You have R5 that you can spend. The best choice would be offer 1 from Gang 1 and offer 2 from Gang 2 yielding a total protection time of 7!

Input (mafia.in)

The first line on the input file contain two numbers, G and M. G is the number of gangs with which you are involved. M is the amount of money you have available to spend. The next G lines of input each represent the offers available from that gang and contain a positive integer O followed by O pairs of integers (I, J) each indicating the price and period of that particular offer.

Sample Input:

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

Output (mafia.out)

The output file should contain a single number indicating the maximum amount of time that you can guarantee your survival.

Sample output:

```
7
```

Constraints

$1 \leq G \leq 100$
$1 \leq O \leq 100$
$0 \leq M \leq 5{,}000$
$0 \leq I, J \leq 1{,}000{,}000$

Time limit

2 seconds.

Scoring

An optimal solution will score 100%, anything else will score 0% and anger the Gods!

Enlightened Landscape

Author

Richard Starfield

Introduction

Consider a landscape composed of connected line segments. N light bulbs hang above the landscape at some height T. Their purpose is to light the entire landscape.

Task

Write a program to output the minimal set of bulbs necessary to light the entire landscape. A landscape point is considered lit if it can “see” a light bulb directly, that is, if the line segment which links the point with a bulb does not contain any other landscape segment’s point. Line segments are considered to include their endpoints.

Example

Suppose a landscape consists of a single horizontal line with two bulbs above it. Either of the bulbs will light the landscape thus a valid optimal solution will use exactly one of these bulbs.

Input (lights.in)

Line 1:
M: An integer, the number of landscape height specifications, including the first and the last point of the landscape.
Lines 2..M+1:
X_i, H_i: Two integers, separated by a space: the landscape height Hi at horizontal position X_i, $1 \leq i \leq M$; for $1 \leq i \leq M-1$ we have $X_{i+1} > X_i$; any two consecutive specified points identify a segment of line in the landscape.
Line M+2:
N T: Two integers, separated by a space, the number of light bulbs and their height coordinate (altitude). The bulbs are numbered from 1 to N)
Line M+3:
B_1 B_2 ... B_N: N integers, separated by spaces: the horizontal coordinates of the light bulbs $B_{i+1} > B_i$, $1 \leq i \leq N-1$;

Sample input:

```
6
1 1
3 3
4 1
7 1
8 3
11 1
4 5
1 5 6 10
```

Output (lights.out)

Line 1:
K: An integer: the minimum number of light bulbs to be switched on.
Line 2:
L_1 L_2 ... L_K: K integers, separated by spaces: the labels of the light bulbs to be switched on specified in increasing order of their horizontal coordinates.
Extraneous white space at points in the output file where white space is permitted will not affect the validity of a solution.

Sample output:

```
2
1 4
```

Constraints

$1 \leq M \leq 200$
$1 \leq N \leq 200$
$1 \leq X_i \leq 10000$ for $1 \leq i \leq M$
$1 < T \leq 10000$
$1 \leq H_i \leq 10000$ for $1 \leq i \leq M$
$T > H_i$ for any $1 \leq i \leq M$
$X_1 \leq B_1$ and $B_N \leq X_M$
The task always has a solution for the test data. If there are multiple solutions, only one is required.

Time limit

1 second.

Scoring

10 points will only be awarded to solutions lighting the entire landscape with the minimum number of bulbs.
If the optimal answer is O and your answer is A>O, then 10*exp(1-(N-O)/(N-A)) rounded down points will be awarded if the entire landscape is lit but unnecessary bulbs are used.
0 points will be awarded to a solution leaving any landscape points unlit.