ACM-ICPC Live Archive

4762 P2P File Sharing System

Peer-to-peer(P2P) computing technology has been widely used on the Internet to exchange data. A lot of P2P file sharing systems exist and gain much popularity in nowadays.

Let's consider a simplified model of P2P file sharing system: There are many computers in the system, and they can receive data from or send data to others. To simplify the problem, let's assume that there is just ONE large file which is of our concern in the system. Some computers already have the whole file (we call them "servers") and some don't(we call them "clients"). Every client needs to download the file from servers. When a client gets the WHOLE file, it becomes a server.

These computers are not always online. An online client will down load the file from all online servers. Different servers send data of different parts of the file to a client, so the client can download the file faster.

Now given the transfer speed between each pair of computers, what time is every computer online or offline, and which computers are servers at the beginning, please analyze the running of the system in a period of time.

Input

The first line contains an integer indicating the number of test cases.

For each test case:

Line 1: It contains two positive integers: n and T ($n \leq 20$, $T \leq 1000$) meaning that there are n computers in the system numbered from 1 to n, and you task is to figure out that how many percentage of the file does every computer gets after T seconds from the beginning.

Line 2: It contains two positive integers: k and S ($k \leq n$, $S \leq 2^{20}$) meaning that at the beginning there are k servers, and the size of the file of our concern is S (KB).

Line 3: It contains k integers. It's a list of all servers' *No.*

Line 4..n+3: These n lines form a matrix of size $n \times n$. The j-th integer in the i-th row of the matrix represents the transfer speed (in KB/s, no more than 2^{10}) between computer i and computer j (i and j start from 1). The matrix is symmetrical and the integers on the diagonal are meaningless.

Line n+4..2n+3: Each line contains an online/offline pattern for a computer in the following format (These lines are in the ascending order of computer *No.*):

$$t\ online\_time_1\ offline\_time_1\ online\_time_2\ offline\_time_2 \ldots online\_time_t\ offline\_time_t$$

t is an integer no more than 10 and the time given are all non-negative integers and in ascending order. During the time between $online\_time_i$ and $offline\_time_i$, the computer is online and can download data from other computers or send data to other computers.

Line 2n+4: It contains one positive integer m, representing the number of download actions in the system.

Line 2n+5..2n+m+4: Each line contains two integers representing a download action in the following format:

$$download\_time_i\ computer\_id_i$$

At time $download\_time_i$, the computer $computer\_id_i$ starts to download the file, where $0 \leq download\_time_i \leq T$, $1 \leq computer\_id_i \leq n$.

These lines are given in non-descending order of time. It's guaranteed that servers never try to download the file. It's ensured that at time $download\_time_i$ the computer $computer\_id_i$ is online (Though it's possible that it instantly go offline after issuing a download command).

When a client starts to download, it will try to connect to all servers and download data simultaneously from online servers. The client's download speed is the sum of all connections. We assume the construction of a connection to be instant and cost no time. Only data transfer is time consuming.

When a client goes offline, unfinished download task will be saved and continued when it's online next time. If a server goes online, all computers that are currently downloading will connect to it and download data from it. What's more, when a client becomes a server, it begins to send data to clients immediately.

NOTE: To simplify the problem, time used to download a file should be rounded up to integer (If the file size is 6KB and download speed is 5KB/s, the download task will cost 2 seconds instead of 1.2 seconds — 5KB for the first second and 1KB for the next second).

Please note that all times given above are in seconds.

Output

For each test case, the output should contain n lines, each for a computer.

The i-th line contains a percentage indicating the amount of data the i-th computer gets after T seconds from the beginning, in the format: '*percentage*%'. The percentage should be rounded down to integer.

Hint:

There are 2 computers. The file size is 1024 KB and Computer 1 is a server at the beginning. At time 10, computer 2 start to download that file from computer 1 until it goes offline at time 40. Totally 10KB/s × 30s = 300KB is downloaded. 300/1024 ≈ 29%.

Sample Input

```
2
2 50
1 1024
1
0 10
10 0
1 0 50
1 10 40
1
10 2
4 500
2 200
2 3
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
2 0 200 300 500
1 100 200
1 200 400
1 301 500
2
0 1
301 4
```

Sample Output

```
100%
29%
100%
100%
100%
99%
```