Problem K
Projects

The 3rd Universal Cup, Stage 40: Potyczki. Limits: 1024 MB, 2 s. *21.06.2025*

You are a manager in a software company. Your n programmers have m projects scheduled for today. Project i can only be worked on by programmers a_i and b_i. Programmer a_i would finish the project in t_i byteseconds, and programmer b_i (if $a_i \neq b_i$) can work twice slower on this project. The two programmers can split the work: programmer a_i can choose a *real* value x_i ($0 \leq x_i \leq t_i$) and spend x_i byteseconds on project i. In that case, programmer b_i needs to spend $2 \cdot (t_i - x_i)$ byteseconds on this project. It may happen that $a_i = b_i$, in which case programmer a_i must complete the entire project on their own, spending t_i byteseconds.

Programmers a_i and b_i can work on project i independently. For example, they can work simultaneously or allocate their time to the project at completely different moments.

Each programmer can work on only one project at a time. The total time a programmer will work is the sum of the times spent on all projects. We assume that switching between projects takes negligible time.

All programmers will start work at the same time today. Some of them will tell you that they are in a hurry and want to leave work as early as possible. As a good manager, you want to meet these expectations. However, you do not yet know exactly which programmers are in a hurry. Therefore, you want to consider q independent scenarios. Each is described by a non-empty subset of programmers who are in a hurry. For each scenario, determine the smallest possible real number T such that you can schedule the work of the programmers so that all projects are eventually completed, and each programmer in a hurry works for no more than T byteseconds.

It can be proven that the results will be rational numbers. So provide all answers as irreducible fractions.

Input

The first line of the input contains three integers n, m and q ($1 \leq n \leq 13$; $1 \leq m \leq 200$; $1 \leq q \leq 10\,000$), representing the number of programmers, the number of projects, and the number of scenarios to consider, respectively.

Each of the next m lines contains three integers a_i, b_i and t_i ($1 \leq a_i, b_i \leq n$; $1 \leq t_i \leq 1\,000\,000$), representing the programmers responsible for a given project and the time required to complete the project by programmer a_i.

The next q lines describe the scenarios; the i-th scenario is one line that contains a binary string s_i of length n; the j-th character is '`1`' if the j-th programmer is in a hurry, and '`0`' otherwise. Each binary string s_i contains at least one character '`1`'.

Output

The output should consist of q lines; the i-th line should contain one rational number T written in irreducible fraction form x/y ($\mathrm{GCD}(x, y) = 1$ and $y > 0$) – the minimal possible limit on the work time for the programmers in a hurry in the i-th scenario.

Example

For the input data:

```
5 7 7
2 1 2
2 2 1
3 2 3
3 4 5
4 3 2
1 5 7
1 5 7
10000
01000
00110
11100
11111
01111
01111
```

the correct result is:

```
0/1
1/1
4/1
18/7
28/3
19/4
19/4
```

Explanation:

In the first scenario, programmer 1 can leave work immediately as the other programmers can handle the projects without him.

In the second scenario, programmer 2 must complete the second project, which will take him 1 bytesecond.

In the third scenario, programmer 4 will spend 2 byteseconds on project five and 2 byteseconds on project four. And programmer 3 should spend 4 byteseconds on project four.

In the fourth scenario, programmers 1 and 3 will each spend $\frac{18}{7}$ byteseconds on projects one and three, respectively. Programmer 2 needs to spend, respectively, $\frac{5}{7}$, 1 and $\frac{6}{7}$ byteseconds to finish the first three projects, finishing the work day at time $\frac{5+7+6}{7} = \frac{18}{7}$ too.

In the fifth scenario, programmers 1 and 5 each need to spend at least $\frac{28}{3}$ byteseconds on the last two projects. There is a strategy where each programmer finishes work after at most $\frac{28}{3}$ byteseconds.