2016 Canadian Computing Olympiad
Day 1, Problem 1
Field Trip

Time Limit: 2 seconds

Problem Description

As a special treat for your kindergarten class, you're taking them on a field trip to a magical place of wonder.

Your class has N students, numbered from 1 to N for convenience. There are M direct, two-way friendships that exist between the students. Each student is friends with at most two other students.

Aside from the M direct friendships, students may also be acquainted with one another. Two students i and j are acquaintances if they're friends, or if there exists a third student k who is an acquaintance of both students i and j. For example, if $(1, 2)$, $(2, 3)$, $(3, 4)$ and $(4, 5)$ are pairs of students with a direct friendship, then person 1 and person 5 are acquaintances.

You're getting ready to order buses for the trip, but there are two issues. Firstly, the transportation company insists that every bus you order must be filled exactly to its capacity of K students. They won't allow you to order a bus if you intend to put fewer than K students on it! Secondly, the students are picky about their travelling conditions. Each student i will refuse to get on a bus unless both of the following conditions are met:

1. All of the other students getting on that bus are acquaintances of student i;
2. All of student i's acquaintances are getting on that bus.

Unfortunately, it looks like you might not be able to bring your whole class on this trip after all. However, you'll do whatever it takes to get as many students as possible on buses. As it turns out, "whatever it takes" may involve putting an end to a friendship or two, for the greater good. You may choose to sever 0 or more of the M friendships amongst the students, which will of course also have an effect on which students are acquainted with one another.

Determine the maximum number of students which can be brought on the trip, such that they're loaded onto buses with exactly K students each, and every student is satisfied with their bus allocation. Furthermore, since you're feeling generous, determine the minimum number of friendships which you can sever in order to be able to bring that many students along.

Input Specification

The first line contains three space-separated integers N, M and K ($1 \leq N \leq 10^6$; $0 \leq M \leq 10^6$; $1 \leq K \leq N$).

The next M lines contain information about the friendships. That is, each of these M lines contain two space-separated integers A_i and B_i ($1 \leq i \leq M$) describing that students A_i and B_i are friends ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$). Note that no friendship is specified twice (that is, no two unordered friendship pairs are equal to one another).

For 3 of the 25 marks available, $N \leq 1000$.

Output Specification
The output consists of two space-separated integers printed on one line. The first integer is the the maximum number of students which can be brought on the trip. The second integer is the minimum number of friendships which must be severed in order to bring that many students.

Sample Input
```
8 5 2
1 4
8 2
4 5
6 2
3 5
```

Output for Sample Input
```
6 2
```

Explanation for Output for Sample Input
If the friendships between student pairs (8,2) and (4,5) are severed, then 3 buses can be filled as follows:

- Bus 1: Students 1 and 4
- Bus 2: Students 2 and 6
- Bus 3: Students 3 and 5

2016 Canadian Computing Olympiad
Day 1, Problem 2
Splitting Hares

Time Limit: 15 seconds

Problem Description

As you know, some bunnies are good bunnies, and some bunnies are bad bunnies.

You are given the location of all the bunnies, and their "goodness" weight (a positive integer for good bunnies and a negative integer for bad bunnies). No two bunnies are at the same location. Divide them into two groups using a straight line such that the sum of the "goodness" of the bunnies on one side of the line is as large as possible. A bunny on the line is counted in the sum of the weights on both sides of the line.

Input Specification

The first line contains N ($2 \leq N \leq 4000$), the number of bunnies. The next N lines contain three space-separated integers: x_i y_i w_i, which indicates that at the point (x_i, y_i) there is a bunny with a goodness weight of w_i ($-1\,000\,000 \leq x_i \leq 1\,000\,000$; $-1\,000\,000 \leq y_i \leq 1\,000\,000$; $-10\,000 \leq w_i \leq 10\,000$). The locations (x_i, y_i) will be distinct (i.e., there is no other $j \neq i$ such that $(x_i, y_i) = (x_j, y_j)$).

For 5 of the 25 marks available, $N \leq 200$ and no three locations are collinear.

For an additional 10 of the 25 marks available, no three locations are collinear.

Output Specification

Output the maximum sum of weights that is possible by drawing a straight line and picking all the bunnies which are on one side of that line.

Sample Input

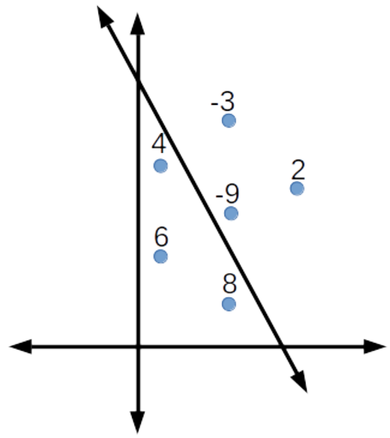
```
6
1 8 4
1 4 6
7 7 2
4 10 -3
4 6 -9
4 2 8
```

Output for Sample Input

```
18
```

Explanation for Output for Sample Input

Take the bunnies with goodness weights of 4, 6 and 8, which are on the “left” side of the line, as shown in the diagram below:



2016 Canadian Computing Olympiad
Day 1, Problem 3
Legends

Time Limit: 2 seconds

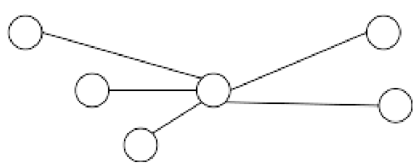
Problem Description

The country of Canada consists of a network of cities and roads. Each road can be traversed in both directions. It is possible to get from any city to any other city using the roads.

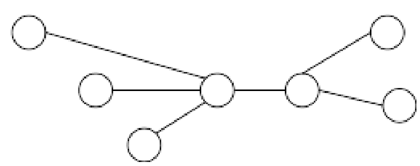
Suzie studies the creation myths of the Canadiaan people. She is particularly interested in five myths (which correspond to the five subtasks of this problem). The myths are very similar. Each myth has the following form:

In the beginning, Canada's road network had a particular structure. As time went on, the network was modified to meet the needs of Canada's growing population. Each modification had one of the following forms:

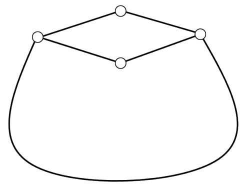
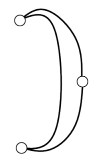
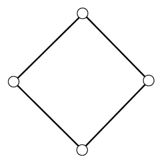
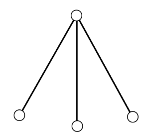
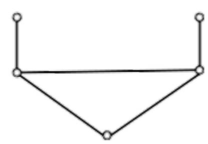
- A road was built between two cities that did not yet have a road going directly between them.
- A new city was built. Cities built in this way were not initially connected to any existing cities.
- A city u grows too large and is split into two cities v and w. The cities originally joined directly to u by a road are partitioned into sets A and B. A road is built from each city in A to v, from each city in B to w and from v to w. For example,



becomes



The five myths only differ in the structure that they believe Canada began with. Here are the original structures, according to each myth:

<table>
<tr><th>Subtask 1 [The Myth of the Flask]</th><th>Subtask 2 [The Myth of the Moon]</th></tr>
<tr><td></td><td></td></tr>
<tr><th>Subtask 3 [The Myth of the Sun]</th><th>Subtask 4 [The Myth of the Eagle's Talon]</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">Subtask 5 [The Myth of the Fox]</th></tr>
<tr><td colspan="2"></td></tr>
</table>

For each subtask, you must take the layout of Canadia as input and determine whether the myth might be correct.

Subtasks are worth 5 marks each.

Input Specification

The first line contains a single integer S ($1 \leq S \leq 5$) representing the subtask which you must solve. The second line contains an integer T ($1 \leq T$) representing the number of test cases. Each test case consists of a blank line, followed by two integers N and M ($2 \leq N$, $1 \leq M$) representing the number of cities and roads, respectively. The cities are numbered from 1 to N. Then M lines follow, each containing two integers a and b ($1 \leq a, b \leq N$) representing two cities connected by a road. No road connects a city to itself. No two roads connect the same pair of cities. It is possible to get from any city to any other city using the roads.

In subtask 3, you may assume that the sum of N over all test cases is at most 10^5. In all other subtasks, the sum of N over all test cases is at most 1000.

The same condition holds for M. In particular, in subtask 3, you may assume that the sum of M over all test cases is at most 10^5. In all other subtasks, the sum of M over all test cases is at most 1000.

Output Specification

For each test case, output a single line containing the string `YES` or the string `NO`.

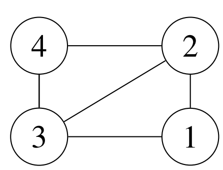
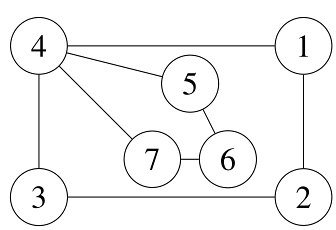
Sample Input 1

```
1
2

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

7 8
1 2
2 3
3 4
4 1
4 5
5 6
6 7
7 4
```

Output for Sample Input 1

```
YES
NO
```

Explanation for Output for Sample Input 1

Test Case Number	Network	Explanation
1		matches The Myth of the Flask
2		cannot match The Myth of the Flask

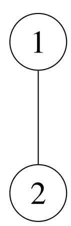
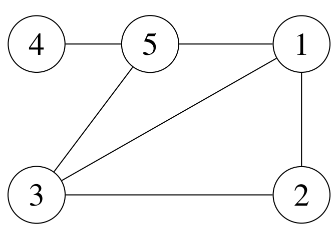
Sample Input 2

```
2
2

2 1
1 2

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

Output for Sample Input 2

```
NO
YES
```

Explanation for Output for Sample Input 2

Test Case Number	Network	Explanation
1		cannot match The Myth of the Moon
2		matches The Myth of the Moon, since we can add cities 4 and 5 along with some extra roads to the Moon formed by cities 1, 2 and 3.

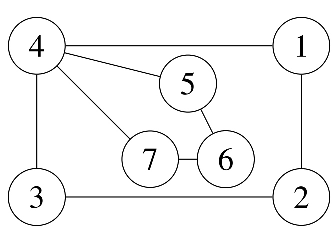
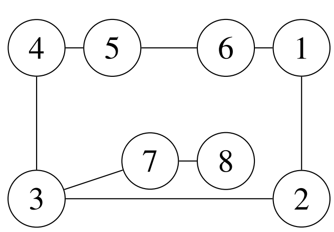
Sample Input 3

```
3
2

7 8
1 2
2 3
3 4
4 1
4 5
5 6
6 7
7 4

8 8
1 2
2 3
3 4
4 5
5 6
6 1
7 3
8 7
```

Output for Sample Input 3

```
YES
YES
```

Explanation of Output for Sample Input 3

Test Case Number	Network	Explanation
1		can match The Myth of the Sun, on cities 1, 2, 3 and 4
2		can match The Myth of the Sun, on cities 1, 2, 3 and 4 where some new cities have been inserted between cities 1 and 4

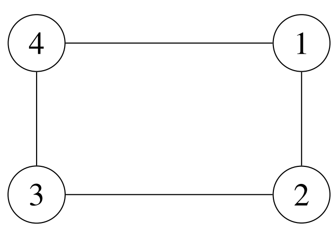
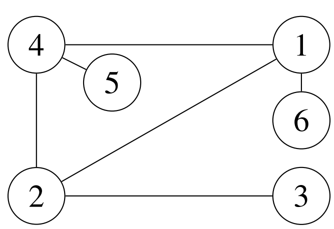
Sample Input 4

```
4
2

4 4
1 2
2 3
3 4
4 1

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

Output for Sample Input 4

```
NO
YES
```

Explanation of Output for Sample Input 4

Test Case Number	Network	Explanation
1		cannot match The Myth of the Talon
2		can match The Myth of the Talon on cities 1, 2, 4 and 6

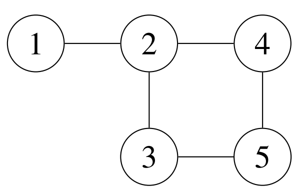
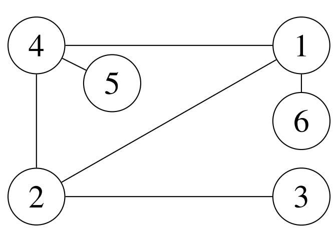
Sample Input 5

```
5
2

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

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

Output for Sample Input 5

```
NO
YES
```

Explanation of Output for Sample Input 5

Test Case Number	Network	Explanation
1		cannot match The Myth of the Fox
2		can match The Myth of the Fox, on cities 1, 2, 4, 5 and 6