

(Adv.) Competitive Programming

Submit until 21.06.2019 13:30, via the [judge interface](#)



Problem: campuses (1 second timelimit)

As the HPI has grown ever larger, its number of campuses has gone beyond the single and even double digit range. With that, the travel time from one lecture to another has become quite the problem. Now the HPI management has decided to try a radical approach: underground railway connections between the campuses.

Each campus consist of a number of rooms, with some connections between adjacent rooms. They were designed with german efficiency in mind, allowing exactly one path between every pair of rooms.

To implement the plan, one or more rooms per campus will be repurposed into railway stations. These will then be connected with high-speed railway routes, where every route goes from one station to another. The management insists that these still only allow a single path between every two rooms, since they consider anything more a waste of money.

The job of planning which routes to build fell on you. Your goal is to minimize the maximum time it takes to get from any room to any other. Since the exact time would be quite hard to know, you just assume that walking from one room to an adjacent one takes a fixed amount of time. The railways will be fast and run frequently, so you just treat each of the railway routes like a room connection.

Input The first line contains n and m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m < n$), the total number of rooms and room connections in all campuses. The next m lines each contain a room connection in the form of a and b ($1 \leq a, b \leq n$, $a \neq b$), where a and b are room numbers. There is at most one connection between every pair of rooms.

Output Output a line containing r , the number of railway routes to add, and t , the maximum time to get from any room to any other, where walking from one room to an adjacent one takes time 1. Then output r lines each containing a railway connection by printing both endpoints. You can output them in any order. If there are multiple solutions, you can print any of them.

Sample input

```
2 0
```

Sample output

```
1 1
1 2
```

4 2

1 2

2 3

1 2

4 2

10 7

1 2

3 2

2 4

5 6

7 8

7 9

10 7

2 4

2 7

5 7