A. Graph Efficiency

Limits: 4 sec., 1024 MiB

Given a connected undirected graph G, you are to add some edges to it in order to maximize the efficiency of the graph.

Let's define $d(v, u)$ as the distance between two vertices of the graph, i.e. the minimum number of edges on a path between two vertices. Then the efficiency of a pair of vertices is the reciprocal of the distance between them:

$$\mathit{Eff}(v, u) = \frac{1}{d(v, u)}.$$

Let's define the efficiency of a vertex v as an average efficiency of v and each other graph vertex:

$$\mathit{Eff}(v) = \frac{1}{n-1} \sum_{u \neq v} \mathit{Eff}(v, u).$$

Where n is the number of vertices of the graph.

Finally, the efficiency of a graph is an average efficiency of all vertices:

$$\mathit{Eff}(G) = \frac{1}{n} \sum \mathit{Eff}(v).$$

You are given a graph G with n vertices and m edges. Your task is to add exactly k edges to the graph in order to maximize the efficiency of the graph. Please note that you are not required to find the optimal answer for each test case. Your score for the test is proportional to $\mathit{Eff}(G')$ where G' is the initial graph G with edges you added. Please refer to the Scoring section for more details.

Input

The first line of the input contains three integers n, m and k: the number of vertices of the graph, the number of edges of the graph and the number of edges that you have to add to the graph respectively. The following m lines describe the edges of the graph. Each of them contains a pair of integers v and u: indices of edge endpoints. Indices of vertices are 1-based. There are no multiple edges or loops, the graph is connected.

Output

Print k lines that describe edges you want to add to the graph. Each line should contain a pair of integers v and u: indices of edge endpoints.

It is guaranteed that it is possible to add k distinct edges to the graph.

You are not allowed to add the same edge twice, add already existing edges, or create loops.

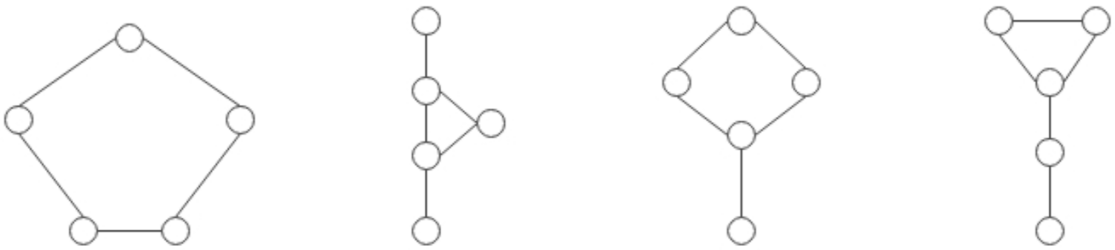
Constraints

$1 \leq n \leq 200$,
$n - 1 \leq m \leq n \cdot (n-1)/2$,
$1 \leq u, v \leq n$,
$1 \leq k \leq 5$.

Samples

Input (*stdin*)	Output (*stdout*)
5 4 1 1 2 2 3 3 4 4 5	1 5

Notes



For example, consider P_5, a path with 5 vertices. There are 4 different P_5 with added edge up to isomorphism. By calculation $\mathit{Eff}(P_5) = 77/120 \approx 0.6417$.

Graphs above are 4 different P_5 with added edge. Lets label them G_1 to G_4 from left to right respectively. Here $\mathit{Eff}(G_1) = 3/4 = 0.75$, $\mathit{Eff}(G_2) = 11/15 \approx 0.733$, $\mathit{Eff}(G_3) = 11/15 \approx 0.733$ and $\mathit{Eff}(G_4) = 43/60 \approx 0.7167$. The first option yields the best result.

Tests

All test cases including provisional and final sets are generated by the generator. Please, check the generator source code and feel free to use it for local testing.

Scoring

If your output for the test is invalid, i.e. the edges you added are incorrect, your score for a test is 0. Otherwise your score for the test is

$$\left\lfloor \mathit{Eff}(G') \cdot 10^7 \right\rfloor .$$

Where $\mathit{Eff}(G')$ is the efficiency of the initial graph with the edges you added.

Your overall score is the sum of your scores over all test cases.

Submissions

- The execution time limit is 4 seconds per test case and the memory limit is 1024 mebibytes.
- The code size limit is 64 kibibytes.
- The compilation time limit is 1 minute.
- There are 50 provisional test cases. Your submissions will be evaluated on the provisional set during the submission phase.
- You can submit your code once per 10 minutes and you will get feedback with your normalized score for each of the provisional tests.

- There will be 500 test cases in the final testing after the submission phase is over. The final results will be announced within two weeks.

Quick start

Check the sample solution which simply finds the edges with the smallest indices that can be added to the graph. The source code is available for some of the contest programming languages:

- C++
- Python
- Java
- C#