Assignment-5

Note: Start working on the following problems. More problems will be added within a day or two.

Problem 1

A network can be said to be vulnerable if there exist nodes whose removal disconnects the network.

Given an undirected graph representing a network, find all such nodes.

Input

T - number of test cases

For each test case,

  V - number of vertices

  E - number of edges

  For each edge,

    v1 v2 - a pair of vertices

Output

For each test case, output the vulnerable nodes in increasing order of their indices. If no such nodes exist, print NONE.

Limits

------

1 <= T <= 100

1 <= V <= 1000

1 <= E <= 10000

Sample Input

3

6 6

0 1

1 2

0 2

2 3

3 4

4 5

7 8

0 1

1 2

2 0

1 3

1 4

1 6

3 5

4 5

3 3

0 1

1 2

2 0

Sample Output

----

2 3 4

1

NONE

Problem 2

A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component (SCC) of a directed graph is a maximal strongly connected subgraph.

Given a graph, find the number of strongly connected components.

Input

T - number of test cases

For each test case,

    V - number of vertices

    E - number of edges

    For each edge,

        v1 v2 - two vertices

Output

For each test case, print the number of strongly connected components followed by a new line.

Limits

----

1 <= T <= 10

1 <= V <= 500

1 <= E <= 10000

Sample Input

2

5 5

1 0

2 1

0 2

0 3

3 4

8 11

0 1

1 2

2 0

2 3

2 5

3 5

5 7

3 4

4 6

3 6

6 3

3 6

Sample Output

3

4

Problem 3

Given a directed acyclic weighted graph, find the longest paths from a source vertex 's' to all other vertices.

Input

T - number of test cases

For each test case,

    V - number of vertices

    E - number of edges

    For each edge,

        v1 v2 w - pair of vertices and the weight

    s - the starting vertex

Output

For each graph, print the longest distances of all the vertices in the ascending order of their indices.

If no path exists between the source vertex and some vertex, print INF.

Limits

------

1 <= T <= 100
1 <= V <= 500
1 <= E <= 10000

Sample Input

2
6 10
0 1 5
0 2 3
1 3 6
1 2 2
2 4 4
2 5 2
2 3 7
3 5 1
3 4 -1
4 5 -2
1
7 8
0 4 10
0 3 10
1 2 40
2 3 60
2 4 25
3 4 30
2 6 35
4 5 40
0

Sample Output

INF 0 2 9 8 10

0 INF INF 10 40 80 INF

Problem 4

Given a graph, find the minimum spanning tree.

Input

T - number of test cases

For each test case,

V - number of vertices

E - number of edges

For each edge,

v1 v2 w - pair of vertices and weight

Output

For each test case, print the vertex pair and the edge-weight separated by spaces and followed by a new line (in increasing order of edge-weights).

Limits

------

1 <= T <= 100

1 <= V <= 5000

1 <= E <= 50000

Sample Input

2

8 8

0 2 71

0 1 54

1 2 91

4 5 80

5 6 5

2 3 25

3 4 39

6 7 72

4 3

1 2 67

0 1 7

2 3 55

7 7

0 2 79

0 1 3

1 2 12

4 5 14

5 6 97

2 3 69

3 4 99

Sample Output

----

5 6 5

2 3 25

3 4 39

0 1 54

0 2 71

6 7 72

4 5 80

0 1 7

2 3 55

1 2 67

0 1 3

1 2 12

4 5 14

2 3 69

5 6 97
3 4 99

Problem 5

Given a Directed Acyclic Graph, find the number of paths from a source vertex S to a destination vertex T.

Input Format:

The first line contains T, the number of test cases. The first line of each test case contains V, the number of vertices. The next line contains S and T. The following line contains E, the number of directed edges, followed by each edge on a single line. No edge will be repeated in a test case.

Constraints:

1<=T<=10
2<=V<=500
1<=E<=V*(V-1)/2
1<=S,T<=V. (S!=T)

Time limit: 1 second

Output Format

Output T lines, one for each test case with the number of paths from S to T (modulus 2^20).

Sample input:

2

7

1 7

11

1 2

1 3

1 4

2 5

2 6

3 5

3 6

4 5

4 6

5 7

6 7

3

1 3

2

1 2

2 3

Sample Output

6

1