Indian Institute of Technology, Mandi
August - November 2019

CS202 - Data Structures and Algorithms

Programming Assignment 4

Course Instructor : Aditya Nigam

Assignment created by : Ranjeet Ranjan Jha

1 October 2019

Instructions

- Plagiarism is strictly prohibited. In case of violation, an F grade will be awarded.
- Submit the complete code as cs202assignment4.zip file with a Makefile.
- You are required to use the template files provided in this assignment. For every problem, these templates are only to be used.
- The deadline for submission is **Friday, 8th November, 2019, 08:00 HRS (8:00 am)**. No late submissions will be entertained.
- Contact Dilip chauhan (7018378273), Ranjeet Ranjan Jha (7066896525) or Daksh Thapar (9592563214) for any queries.

1 Implementation

- You are given 6 .hpp files. Each of the problems have to be solved using these files only.
- For each problem there will be a separate main file but every problem will share the .hpp files.
- For each problem, the code should save the inherent graph as adjacency list in a file named ‘adj lst.txt’ and adjacency matrix in a file named ‘adj mat.txt’. The code should be capable of also taking the input directly from the adjacenecy list and matrix rather than the specified format defined for that problem.

  10 marks are reserved for proper implementation and will only be awarded when all the above points are implemented for each problem. **10 marks**

2 Problems

1. Given a rectangle with left bottom coordinate as (0, 0) and right up coordinate as (x, y). There are K circles with centers inside the rectangle.Radius of each circle is R.Now we need to find out if it is possible that we can move from (0, 0) to (x, y) without touching any circle.
   Note: We can move from any cell to any of its 8 adjacent neighbours and we cannot move outside the boundary of the rectangle at any point of time.

   Input Format

   1st line of input contains an integer T ,the total number of test cases.
   Each test case contains 5 lines of input.
   1st line contains two space separated integers x and y ,the right up coordinate.
   2nd line contains an Integer K ,the number of circles.
   3rd line contains an Integer R ,the radius of each circle.
   4th line contains K space separated integers where ith integer denote x coordinate of the ith circle.
   5th line contains K space separated integers where ith integer denote y coordinate of the ith circle.

   Output Format
   For each test case print in a new line “YES” or “NO” depending on weather it is possible to reach cell (x,y) or not starting from (0,0).

Constraints

$0 < x, y, R < 100$

$1 < N < 1000$

Center of each circle would lie within the grid

Examples

Input:

1

2 3

1

1

2

3

Output:

NO

Explanation for test case 1:

There is NO valid path in this case

2. Given a N x M binary matrix A of 0's and 1's, where 0 = black, 1 = white.Return the number of black shapes. Where a black shape consists of one or more adjacent 0's (diagonals not included).

Input Format

1st line contains an integer T, the number of test cases.

For each test case

1st line contains two space separated integers N the number of rows and M the number of columns.

Next N lines contains M space separated values from set $\{0,1\}$.

Output Format

For each test case print in a new line the number of black shapes.If no black shapes found print 0.

Constraints

$1 <= N,M <= 1000$

Examples

Input:

1

3 7
1 1 1 0 1 1 1
1 1 0 0 1 0 1
1 0 1 1 1 0 1

Output:
3
Explanation for test case 1:
3 black shapes are :
(i) 0
0 0
(ii) 0
(iii) 0
0

3. There are N network nodes, labelled from 1 to N.Given times, a list of travel times as directed edges times[i] = (u, v, w), where u is the source node, v is the target node, and w is the time it takes for a signal to travel from source to target.Now, we send a signal from a certain node K. How long will it take for all nodes to receive the signal? If it is impossible, print -1.

Input Format

1st line contains an integer T, the number of test cases.
For each test case
1st Line contains an integer N ,the number of network nodes.
2nd Line contains an integer K ,the source node.
Next N lines contains 3 space separated integers u ,v and w where u is the source node, v is the target node, and w is the time it takes for a signal to travel from source to target.

Output Format

For each test case print in a new line an integer denoting time taken to receive the signals by all the other nodes if possible otherwise print -1 .

Constraints

N will be in the range [1, 100].
K will be in the range [1, N].
The length of times will be in the range [1, 6000].
All edges (u, v, w) will have 1 <= u, v <= N and 0 <= w <= 100

Examples

Input:

1

4

2

2 1 1

2 3 1

3 4 1

Output:

2

Explanation for test case 1:

In 1st unit of time the signal will reach to node 1 and 3 and in 2nd unit of time the signal will reach to node 4.so in total 2 unit of time the signal will reach to all the other network nodes.

4. There are n cities numbered from 1 to n and there are m roads connecting these cities. Somehow all the roads are damaged simultaneously.Every road has four parameters u,v,w and c where u and v are the two cities connected by the road ,w is the distance between these two cities and c is per unit cost to repair this road.The task is to connect the cities in such a way that every city is reachable from every other city and total repair cost should be minimum.

   Note: It is sure that all the cities were connected before the roads were damaged.Also all the roads are bidirectional.

   Input Format

   1st line of input contains an integer T, the number of test cases.

   For each test case

   1st line contains an integer n denoting the number of cities.

   2nd line contains an integer m denoting the number of roads.

   Next m lines contains 4 space separated integers u ,v ,w and c where u and v are the two cities connected by the road ,w is the distance of this road and c is per unit cost to repair this road.

   Output Format

   For each test case print in a new line an integer denoting minimum cost to connect all the cities.

   Constraints

2<=n<=100
n-1<= m <= n*(n-1)/2

Examples
Input:
1
5
7
1 2 1 1
1 3 2 1
1 4 3 1
1 5 4 1
2 3 5 1
2 5 7 1
3 4 6 1

Output:
10
Explanation for test case 1:
We need to repair the roads (1 2 1 1),(1 3 2 1), (1 4 3 1), (1 5 4 1) and the cost is 1*1+2*1+3*1+4*1=10.

5. **(Optional)** There are a total of n courses you have to take, labeled from 0 to n-1. Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1] Given the total number of courses and a list of prerequisite pairs.is it possible for you to finish all courses? If possible, return the ordering of courses you should take to finish all courses.There may be multiple correct orders, you just need to print one of them

Input Format
1st line of the input contains an integer T, the number of test cases.
For each test case
1st line contains an integer n ,the number of courses.
Next n lines contains two space separated integers u and v where u is prerequisite of course v.

Output Format
For every test case If possible for you to finish all courses then print in a new line the ordering of courses you should take to finish all courses otherwise print “NO”

Constraints

2<=n<=100

0<= u,v<=n-1

Examples

Input:

1

4

1 0

2 0

3 1

3 2

Output:

0 1 2 3 or 0 2 1 3

Explanation for test case1 : There are a total of 4 courses to take. To take course 3 you should have finished both courses 1 and 2. Both courses 1 and 2 should be taken after you finished course 0. So one correct course order is [0,1,2,3]. Another correct ordering is [0,2,1,3].