Digital Pakistan Speed Programing Competition Online Qualifier Round

Instructions

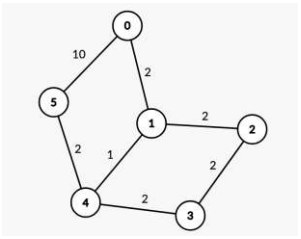
- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, seek a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- All your programs must meet the time constraint specified.
- The decision of judges will be absolutely final.

Problem 01: COVID Vaccination Center Placement

Time Limit: 1 sec

In response to a COVID-19 outbreak, the Ministry of Health is deploying **K** mobile vaccination units across a network of towns to ensure rapid and equitable vaccine access. These units must be strategically placed so that every town has access to at least one center within a reasonable travel distance.

The nation's geography is represented as an undirected, connected graph consisting of **N** towns and **M** bidirectional roads. Each road has a positive integer length, indicating the time required to travel between towns. Travel is symmetrical and uninterrupted.



To optimize deployment, the ministry wants to ensure that no town is too far from its nearest vaccination center. Your task is to choose **K** towns to host the mobile vaccination units such that the maximum shortest distance any town must travel to reach its nearest center is minimized.

Input Format

- The first line contains three integers: **N** ($1 \le N \le 300$), the number of towns **M** ($0 \le M \le N(N-1)/2$), the number of roads, **K** ($1 \le K \le N$), the number of vaccination centers to deploy
- The next **M** lines each contain three integers: **u, v, w** ($0 \le u, v < N$, $1 \le w \le 10^4$), representing a bidirectional road of length **w** between towns **u** and **v**.

Output Format

- Output a single integer:
  The minimum possible value of the maximum shortest distance from any town to its nearest vaccination center, assuming the **K** centers are optimally placed.

Sample Input	Sample Output
6 7 1 0 1 2 1 2 2 2 3 2 3 4 2 4 5 2 0 5 10 1 4 1	3