1101 – A Secret Mission

All of you have heard about Evil Jack Diablo, the one who had stolen the whole problem set from the Good Judges last time. Once again he is making evil plans, but he does not know that Alice is on a secret mission. There will be several pairs of City of Evils on the way of Alice's mission.

There will be **N** evil cities (numbered by **1, 2, ..., N)** connected by **M** bidirectional roads. There will be evil guards patrolling the roads. Since they are not much intelligent, the danger of travelling in each road is not the same. Alice is going to travel from city **s** to city **t**. You can safely assume that it's possible to travel from any city to another. John the legend programmer has estimated the danger of each road but since he is busy arranging contests, Alice is depending on you now. Danger of a path from city **s** to city **t** is defined as the maximum danger of any road on this path. As you are one of the most talented programmers, you will love to help Alice by finding the least dangerous paths to prevent Evil Jack from doing harms to the Judges.

Input

Input starts with an integer **T (≤ 3)**, denoting the number of test cases.

Each case contains two integers **N, M (2 ≤ N ≤ 50000, 1 ≤ M ≤ 10^5)** denoting the number of cities and roads. Each of the next **M** lines contains three integers: $\mathbf{x_i, y_i, d_i}$ **(1 ≤** $\mathbf{x_i, y_i}$ **≤ N,** $\mathbf{x_i \neq y_i}$**, 1 ≤** $\mathbf{d_i}$ **≤** $\mathbf{10^4}$**)** - the cities connected by the $\mathbf{i^{th}}$ road and its degree of danger. Next line contains an integer **q (1 ≤ q ≤ 50000)**. Each of the next **q** lines contains two integers: $\mathbf{s_i}$ and $\mathbf{t_i}$ **(1 ≤** $\mathbf{s_i, t_i}$ **≤ N,** $\mathbf{s_i \neq t_i}$**)**.

Output

For each test case, print the case number first. Then for each query $\mathbf{s_i}$ $\mathbf{t_i}$, print the least dangerous path in a line.

Sample Input	Output for Sample Input
2 4 5 1 2 10 1 3 20 1 4 100 2 4 30 3 4 10 2 1 4 4 1 2 1 1 2 100 1 1 2	Case 1: 20 20 Case 2: 100

Notes

Dataset is huge, use faster i/o methods.