F Fantastic Network

An undirected weighted graph ***G = (V, E)*** is defined as a **Fantastic** network if it has the following properties:

1. The graph is connected.
2. The ***degree*** of any node is at most **6**.
3. It may or may not contain ***cycles***, but the ***length*** of any ***cycle*** (if exists) in this network will be **3**. The nodes which are part of at least one ***cycle*** are called ***fine*** nodes.
4. The ***degree*** of any ***fine*** node can be at most **3**.

Here, ***cycle*** is defined as a path $\langle v_0, v_1, \ldots, v_k \rangle$ in any graph such that the following statements hold:

1. **k** >= **3**. (**k** is the *length* of the *cycle*)
2. $v_0 = v_k$.
3. For each **i** ($0 \leq i < K$) v_i and v_{i+1} are connected by an edge.
4. $v_1, \ldots, v_k$ are distinct.

An ***edge dominating set*** for an undirected graph ***G = (V, E)*** is a subset ***F*** of ***E*** such that every edge not included in ***F*** is adjacent to (i.e. shares a vertex with) some edge in ***F***. The ***weight*** of an ***edge dominating set*** is the sum of the weights of all edges in that set. Given a **Fantastic** network with positive edge weights, you need to determine the weight of the ***minimum weight edge dominating set***.

Input

First line of the input contains a positive integer **T** ($T \leq 100$). The first line of each of the **T** cases contains two integers **N** ($2 \leq N \leq 5000$) and **M** ($1 \leq M \leq 2*N$), representing the number of nodes and edges, respectively, in a Fantastic network. Each of the following **M** lines contains 3 integers u_i, v_i, w_i, which means there is an edge from u_i to v_i ($1 \leq u_i, v_i \leq n$) with weight w_i ($1 \leq w_i \leq 10000$).

Output

For each case, print a line of the form **Case <x>: <y>**, where **x** is the case number and **y** is the weight of ***minimum weight edge dominating set*** of the given Fantastic network.

Sample Input	Output for Sample Input
2 3 2 1 2 1 1 3 2 7 6 1 2 2 1 3 1 2 4 2 2 5 1 3 6 2 3 7 1	Case 1: 1 Case 2: 2

Problemsetter: Anindya Das

Special Thanks: Md. Shiplu Hawlader