Ant Tunneling

Filename: `ant`

The evil ants of UCF have taken your teacher hostage in order to force you to help them with a problem. They are trying to interconnect their ant hills, so that there is some way to get from any ant hill to any other ant hill using only tunnels they have dug. They are quite smart ants, so they have already mapped out all the possible tunnels they can dig to connect their ant hills and assigned each a cost. However, before they get to digging, they demand your help to determine which set of tunnels they should choose to dig to minimize the total cost while still connecting all of their ant hills (directly or indirectly). A pair of ant hills is considered connected if there exists a series of tunnels connected end-to-end that form a path between each of the hills. The ants can start digging from any ant hill, but in the end all pairs of hills must have a path of fully dug tunnels connecting them.

The Problem:

Given the cost of all possible tunnels to connect pairs of ant hills, determine the minimum total cost to dig a tunnel network that connects all ant hills.

The Input:

The first line will contain a positive integer, *n*, the number of campuses for which the ants want you to solve this problem (UCF has many satellite campuses). Following will be descriptions of *n* campuses. Each campus description starts with a line containing two integers, h ($1 \leq h \leq 100$) and t ($0 \leq t \leq h * (h - 1) / 2$), representing the number of ant hills and the number of tunnels the ants can possibly dig, respectively. Each ant hill is assigned a unique number from 1 to h. The following *t* lines will describe the possible tunnels. Each tunnel is described by three integers, x, y ($1 \leq x < y \leq h$) and d ($1 \leq d \leq 1,000$), representing the two ant hills this tunnel connects and the cost of digging this tunnel to connect them, respectively. For each campus, each possible pair of ant hills will have at most one tunnel between them.

The Output:

For each campus, the output should start with "`Campus #c: `" where *c* is the number of the campus in the input (starting at 1). Follow this with either a single integer which is the minimum sum of difficulty costs to dig tunnels in a matter that connects all ant hills (directly or indirectly) or with "`I'm a programmer, not a miracle worker!`" if there is no way to dig tunnels to connect them all.

Sample Input:

```
3
2 1
1 2 5
3 1
1 2 2
4 4
1 2 2
2 3 3
3 4 1
1 4 4
```

Sample Output:

```
Campus #1: 5
Campus #2: I'm a programmer, not a miracle worker!
Campus #3: 6
```