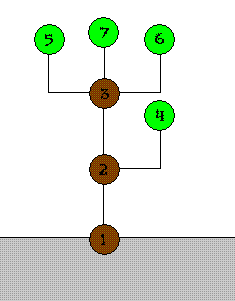
**Big Christmas Tree**

|  |  |  |
| --- | --- | --- |
| **Time Limit:** 3000MS |  | **Memory Limit:** 131072K |
|  |  |  |

**Description**

Christmas is coming to KCM city. Suby the loyal civilian in KCM city is preparing a big neat Christmas tree. The simple structure of the tree is shown in right picture.

The tree can be represented as a collection of numbered nodes and some edges. The nodes are numbered 1 through *n*. The root is always numbered 1. Every node in the tree has its weight. The weights can be different from each other. Also the shape of every available edge between two nodes is different, so the unit price of each edge is different. Because of a technical difficulty, price of an edge will be (sum of weights of all descendant nodes) × (unit price of the edge).

Suby wants to minimize the cost of whole tree among all possible choices. Also he wants to use all nodes because he wants a large tree. So he decided to ask you for helping solve this task by find the minimum cost.

**Input**

The input consists of *T* test cases. The number of test cases *T* is given in the first line of the input file. Each test case consists of several lines. Two numbers *v*, *e* (0 ≤ *v*, *e* ≤ 50000) are given in the first line of each test case. On the next line, *v* positive integers *wi* indicating the weights of *v* nodes are given in one line. On the following *e* lines, each line contain three positive integers *a*, *b*, *c* indicating the edge which is able to connect two nodes *a* and *b*, and unit price *c*.

All numbers in input are less than 216.

**Output**

For each test case, output an integer indicating the minimum possible cost for the tree in one line. If there is no way to build a Christmas tree, print “No Answer” in one line.

**Sample Input**

2

2 1

1 1

1 2 15

7 7

200 10 20 30 40 50 60

1 2 1

2 3 3

2 4 2

3 5 4

3 7 2

3 6 3

1 5 9

**Sample Output**

15

1210

**Source**

[POJ Monthly--2006.09.29](http://poj.org/searchproblem?field=source&key=POJ+Monthly--2006.09.29), Kim, Chan Min (kcm1700@POJ)