

Never have rumours been so fast as in the internet era. Although not as fast, in the old west, they still runned pretty darn fast. As soon as a rumour started in one town it would quickly spread to its neighbours in a matter of hours. Besides that rumours could easily destroy many kinds of deals, both legal and illegal.

This was the context that allowed Lightning Billy to make his living. Billy had a business where he sold to his customers the usefull service of stopping rumours. When a customer came to Billy he would give Billy the details of the rumour being spread. This allowed Billy to estimate how much time he would need to stop that rumour in any given town. With this information and knowing all the possible routes the rumour could take, Billy could estimate in how many towns he could kill the rumour before it reached them (See Figure 1).

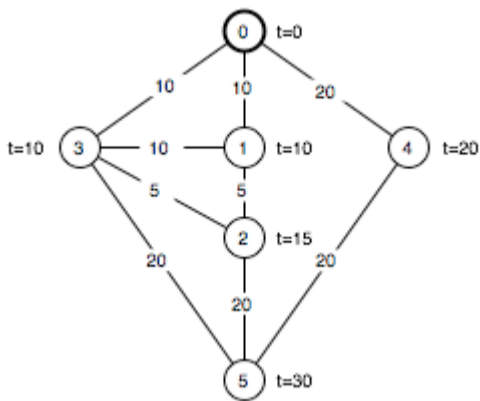


Figure 1 - Towns, routes and rumour distances

Your problem will be to design an algorithm that given a set of towns and routes between them will calculate how many cities Billy can reach in time, knowing that Billy could travel twice as fast as any rumour (he wasn't called Lighthning Billy for nothing). The time needed by Billy to travel through any route will always be rounded down (yes, he really was that fast). Billy will be able to kill the rumour only if he manages to reach a town at least at the same time the rumour gets there.

Input

The input file contains several test cases, each of them as describes below.

The first line of the input will contain the number of towns in the area ($1 \leq nt \leq 50$), and the number of routes ($0 \leq nr \leq 1000$), between them. Then, a single line containing nt values with the time needed by Billy to stop the rumour in each town ($0 \leq s \leq 10$), followed by nr lines containing routes. Each route will be composed by the starting city, the ending city and the time needed for the rumour to spread along that path ($1 \leq t \leq 200$). Note that rumours run both ways between the cities of the route at the same speed, and so does Billy (actually, at half the speed of the rumours, as described above).

Output

For each test case, output a single line containing the number of towns where Billy can stop the rumour. Billy always starts from the first town in the input file. Towns that are not reached by the rumour should not be counted as they are not saved by Billy.

Output explanation: Billy loses 1 time unit to solve the rumour at town 0. He then travels to town 1 taking 5 time units ($10/2$) and arriving there at time unit 6. He loses 2 time units there solving the rumour and then travels to town 2 taking 2 time units and arriving at time unit 10. He loses 3 time units at this town, departing at time unit 13 to town 5 where he arrives at time unit 23 with plenty of time to solve the rumour in that town (See Figure 2). No other towns can be kept safe from the rumour.

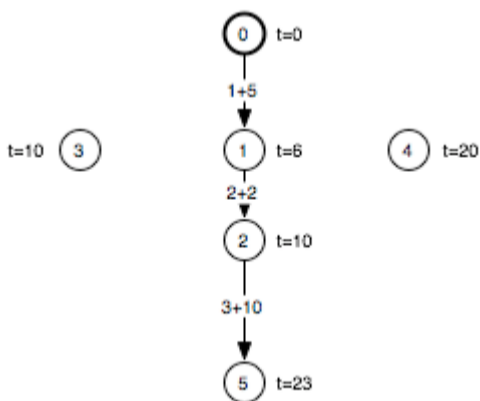


Figure 2 - Output Explanation

Sample Input

```
6 9
1 2 3 4 5 6
0 1 10
0 3 10
0 4 20
1 3 10
1 2 5
2 3 5
2 5 20
3 5 20
4 5 20
```

Sample Output

4