Finances

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Tomorrow (November 17th) we celebrate Debt-Free Day. On this occasion, n friends decided to finally settle their mutual debts from many past trips. They always recorded their expenses in an app, which now summarized and simplified everything. For each person, it displays a total balance a_i – positive if that person is owed money, or negative if that person must repay money. The sum of all displayed values is, of course, equal to zero: $\sum_{i=1}^{n} a_i = 0$. We want to plan transfers (with integer values) in such a way that each person's balance becomes zero.

The n people are connected by m trust relationships, described by triplets (u_j, v_j, c_j) – persons u_j and v_j trust each other enough that either of them can send the other one transfer of value at most c_j. People not directly connected by a trust relationship cannot send money to each other. Unordered pairs $\{u_j, v_j\}$ do not repeat, and the entire trust network is connected – every pair of people is linked by a (possibly multi-step) path of trust relationships.

The app displays the total debt A (equivalently, the sum of positive a_i values). In this group of friends, the trust levels are quite high – for every pair connected by a trust relationship, it holds that $c_j \geq \lceil \frac{A}{2} \rceil$. For example, for the sequence $a = (10, -5, 0, 3, -4, -4)$ we have $A = 5 + 4 + 4 = 10 + 3 = 13$, hence $c_j \geq 7$.

Can this group of n people agree on directions and amounts of transfers to fully settle all debts – that is, make everyone's balance zero? Print `TAK` (yes) or `NIE` (no) for each of the t independent test cases.

You may assume that every person has enough money to perform any necessary transfers.

Input

The first line contains an integer t ($1 \leq t \leq 10^5$), denoting the number of test cases.

The first line of each test case contains two integers n and m ($2 \leq n \leq 10^6$; $n-1 \leq m \leq 10^6$), representing respectively the number of people and the number of trust relationships.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$; $\sum_{i=1}^{n} a_i = 0$), representing the initial balance of each person. At least one of the numbers a_i is nonzero.

We define $A = \sum_{i=1}^{n} \max(0, a_i)$. From the previous condition it follows that $A \geq 1$.

The next m lines describe the trust relationships; the j-th line contains three integers u_j, v_j, and c_j ($1 \leq u_j, v_j \leq n$; $u_j \neq v_j$; $\lceil \frac{A}{2} \rceil \leq c_j \leq 10^{15}$). The network of n people and m trust relationships is connected, and each unordered pair $\{u, v\}$ appears in a given test case at most once.

The sum of all n values across test cases does not exceed 10^6; likewise, the sum of all m values does not exceed 10^6.

Output

Print t lines – for each test case, output the word `TAK` if it is possible to balance all accounts, or the word `NIE` otherwise.

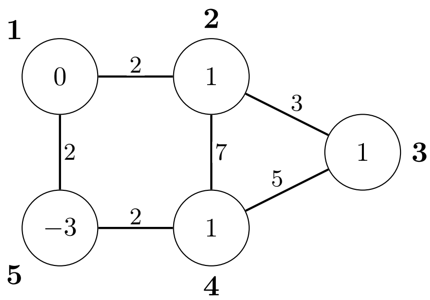
Example

standard input	standard output
3 2 1 20 -20 1 2 10 5 6 0 1 1 1 -3 1 2 2 2 3 3 3 4 5 2 4 7 1 5 2 4 5 2 3 2 1 2 -3 1 2 1000000000000000 2 3 999997235527681	NIE TAK TAK

Note

In the first test case, there are two people with balances $a = (20, -20)$. The first person should send the second one a transfer of value 20, but can send at most $c_1 = 10$. The answer is `NIE` (no).

The figure below illustrates the second test case with balances $a = (0, 1, 1, 1, -3)$ written inside the vertices, and limits c_j on the edges.



Below are two example valid transfer plans. In both plans, transfer limits are not exceeded, and everyone ends up with balance 0. The answer is `TAK` (yes).