# Exercise – *Astérix in Switzerland*

The Roman Empire is falling apart. Many provinces of the empire have a large deficit and are on the verge of declaring bankruptcy. Quaestor Vexatius Sinusitus travels to Gaul to investigate where all the tax money is disappearing to. Varius Flavus, the local governor, is frustrated to learn that Sinusitus does not respond well to bribery and decides to poison the food of the quaestor.

With his life in danger, Sinusitus approaches the druid Panoramix for an antidote to the poison. Panoramix quickly agrees to brew the antidote but is missing one ingredient: edelweiss. Astérix and Obélix embark on a trip to Helvetia to get this rare flower.

With the two heroes gone and the quaestor being sick, you are tasked to save the Roman empire before it is too late! Your job is to decide whether there are parts of the empire worth saving.

There are $n$ provinces in the Roman Empire. Each province $p_i$ has a central bank with a balance $b_i$ that can be either positive (surplus) or negative (deficit). Moreover, some of the provinces are in debt to others. That is, $d_{i,j}$ is the amount of money that province $p_i$ owes to province $p_j$. One or several provinces want to form a new union, so that the debts among them can be canceled. A union of provinces is called free-standing if their total balance minus the total debt to provinces outside of the union is positive. Formally, a union of provinces $X$ is *free-standing* if $\sum_{i \in X} b_i > \sum_{i \in X, j \notin X} d_{i,j}$.

**Input** The first line of the input contains the number $t \leqslant 30$ of test cases. Each of the $t$ test cases is described as follows:

- It starts with a single line containing two integers `n m`, separated by a space. They denote
  - $n$, the number of provinces $p_0, p_1, \dots, p_{n-1}$ ($1 \leqslant n \leqslant 10^3$).
  - $m$, the number of debt relations between provinces ($0 \leqslant m \leqslant 10^5$).
- The next $n$ lines each give the balance of one province. The $i$-th such line contains one integer $\texttt{b}_i$, the balance of province $p_i$ ($|b_i| \leqslant 2^{20}$).
- The next $m$ lines each give one debt relation. Each debt relation contains three integers, $\texttt{i j d}_{i,j}$, separated by a space, and such that $i \neq j$ ($0 \leqslant i \leqslant n-1$, $0 \leqslant j \leqslant n-1$ and $0 < d_{i,j} \leqslant 2^{20}$). This means that province $p_i$ owes $d_{i,j}$ to province $p_j$. Any fixed (ordered) pair $(i, j)$ appears at most once among those lines.

**Output** For every test case output a single line with the string "`yes`" if there is a union of provinces that is free-standing; otherwise, the output string is "`no`".

**Points** There are four groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 20 points, you may assume that there are no more than 20 provinces and 50 debt relations ($n \leqslant 20$ and $m \leqslant 50$).

2. For the second group of test sets, worth 20 points, you may assume that for all $i$ and $j$ we have $d_{i,j} = d_{j,i} > \sum_{k=0}^{n-1} |b_k|$.
3. For the third group of test sets, worth 20 points, you may assume that for all $i$ and $j$ we have $d_{i,j} > \sum_{k=0}^{n-1} |b_k|$, and that all but one province have a negative balance.
4. For the fourth group of test sets, worth 40 points, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3, 4\}$.

### Sample Input

```
2
3 4
3
2
-6
0 2 2
1 0 4
1 2 2
0 1 1
3 3
2
4
-6
0 1 8
1 2 9
2 0 7
```

### Sample Output

```
yes
no
```

看了一眼idea, 感觉我是想不出来...

link all  positive b with source and negative with sink, and debt as positive capacity edges

if the maximum flow is not equal to the sum of all positive balance, then we trace the reverse capacity, and get the set where source can connect to, these subset have non-empty balance, but still, and outedge is full (because the minimum cut - max flow theorem) then we have \sum b_i > \sum (b_i - net_i) = \sum d_{ij}.