
ACM Programming Challenges Lab

Exercise 1 – *Political Compromise*

On a remote country composed of dozens of islands there are two political parties. Exactly half of the population supports the WeWantBest (WWB) party, while the other half supports the Go-ForCheap (GFC) party. A truly unfortunate situation, as they rarely ever agree on something and without a strict majority nothing gets done.

Since both parties agree that roads are a good thing, on the project of building a new highway network they have reached a compromise. The GFC accepts that new highways connecting all cities are built while on the other hand the WWB accepts that the network will have a tree shape with no redundant roads.

Unfortunately the implementation of this compromise proved difficult: the parliament commissioned a report detailing the expected travel time between many cities and the corresponding cost of the highway. Due to mountains and other obstacles however, the fastest road network is not always the cheapest. Only when they are the same the network is built. Your task is to determine on which island roads get built and which will still not get them.

Input The first input line contains the number n of islands. Then n blocks follow with the following structure. The first line of a block contains integers v and e , the number of cities on the island and the number of roads in the report. The next e lines consist each of 4 integers. The first two are between 0 and $v - 1$ and identify the cities at both endpoints of the road. The next two are the travel time in minutes and the cost of the road. You can assume that no two travel times are the same and that no two roads cost the same. The input graph is also always connected.

Output For each island output either “yes” or “no”, depending on whether the road network will be built or not. On any island the network is built if the cheapest cycle-free road network connecting all cities of the island is also the one with the smallest sum of travel times.

Sample input

```
3
2 1
0 1 10 20
3 3
0 1 10 20
1 2 20 10
2 0 30 30
3 3
0 1 10 40
1 2 20 20
2 0 40 10
```

Sample output

```
yes
yes
no
```