Building Road

Time Limit 1 sec/Memory Limit 256 MB

Yomo, as a country of transport minister, needs to build roads for cities. There are n cities numbered from 1 to n in this country. Initially, there were no roads in this country. He wants to make these cities connected. That means for every pair of cities, there exists at least one path that can go through. Every city has a building cost a_i, If we want to build a road for city u and city v. The cost of building a road between cities u and v is $a_u + a_v$ coins.
There are also m special offers, each denoted by three numbers x, y, and w, which means he can connect city x and y by paying w coins for building a road. We don't have to use special offers. If there is a pair of cities x and y that has a special offer, we can pay $a_x + a_y$ coins for building a road to connect two cities.
Yomo has given you the plan, the a_i for all cities and m special offers. What is the minimum number of coins to make the graph connected? You have Q queries.

Input Format

- The first line of the input contains one integer Q which means that Q plans need to check.
- For every query of plan, the first line contains two integers n, m which means that the number of the cities in the country and the number of special offers.
- The second line contains n integers $a_1, a_2, \ldots, a_n$ which means the building cost for city i.
- And the following m lines contains two integers u, v, w, the special offer. Connecting city x and y by paying w coins for building a road.

Output format

- Output the minimum number of coins for every plan.

隱私權 - 條

Constraints

- $1 \leq Q \leq 20$
- $1 \leq n \leq 10^5$
- $0 \leq m \leq 10^5$
- $1 \leq u, v, w \leq n \ (u \neq v)$
- $0 \leq a_i, w \leq 10^9 \ (1 \leq i \leq n)$
- All number are intergers

Credits

- 70% : $n, m \leq 500$
- 30% : no restriction

sample input #1	sample output #1
3 6 1 1 2 1 2 2 2 3 5 1 6 1 2 1 1 2 1 2 2 4 1 6 0 2 2 2 1 2 2	12 11 15

sample input #2	sample output #2
3 6 3 4 3 5 3 5 4 1 5 4 2 5 4 5 6 4 6 3 4 5 5 3 3 3 1 4 5 2 3 5 4 6 4 6 2 4 5 5 4 3 3 3 6 7 4 6 7	26 28 35