

(Adv.) Competitive Programming

Submit until 26.04.2019 13:30, via the [judge interface](#)



Problem: cycling (1 second timelimit)

A small country worried about the increasing congestion of its road networks, had a great idea to get cars off the road. They built a cycling network connecting all the cities in the whole country. However, there is a small problem: The country has lots of mountains. And most cyclists prefer not to cycle up mountains, because it is just too exhausting. So now, years after its construction, the expensive cycling network remains largely unused. The government wants to change that, so they tasked a company to convert some of paths to completely flat ones with an elevator at the end, to cover the distance in heights between the cities. Making the paths flat is easy enough, because the company is great at building tunnels and bridges. Elevators however are expensive to build, since they are taxed proportional to their height, thanks to the strong lobby for stairs in the country. Now your goal is to figure out which existing paths to convert, so that the tax bill for all the elevators is minimal. But of course all cities should be reachable, with the new upgraded paths only.

Input First, there is a line with $1 \leq n \leq 1000$, the number of cities and m , the number of existing paths. The next line contains the elevation $-100 \leq e_i \leq 5000$ in meters for each of the n cities. The following m lines each describe a cycling path: They contain the indices $0 \leq a, b < n$ of the connected cities.

Output Output the smallest possible sum of the elevator heights.

Sample input

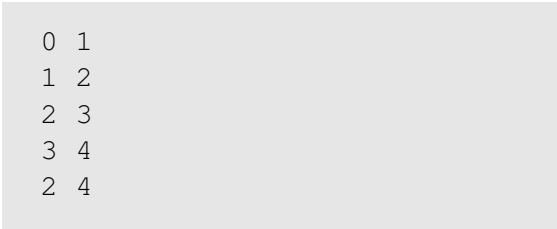
```
3 3
0 1 2
0 1
1 2
2 1
```

```
5 5
10 0 10 20 -10
```

Sample output

```
2
```

```
50
```



0 1
1 2
2 3
3 4
2 4