D. Fewer Batteries

time limit per test: 3 seconds
memory limit per test: 256 megabytes

In 2077, when robots took over the world, they decided to compete in the following game.

There are n checkpoints, and the i-th checkpoint contains b_i batteries. Initially, the Robot starts at the 1-st checkpoint with no batteries and must reach the n-th checkpoint.

There are a total of m one-way passages between the checkpoints. The i-th passage allows movement from point s_i to point t_i ($s_i < t_i$), but not the other way. Additionally, the i-th passage can only be used if the robot has at least w_i charged batteries; otherwise, it will run out of power on the way.

When the robot arrives at point v, it can additionally take any number of batteries from 0 to b_v, inclusive. Moreover, it always carries all previously collected batteries, and at each checkpoint, it recharges all previously collected batteries.

Find the minimum number of batteries that the robot can have at the end of the journey, or report that it is impossible to reach from the first checkpoint to the last.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n, m ($2 \le n \le 2 \cdot 10^5, 0 \le m \le 3 \cdot 10^5$) — the number of checkpoints and the number of passages, respectively.

The second line contains n numbers b_i ($0 \le b_i \le 10^9$) — the number of batteries at the i-th checkpoint.

The next m lines contain three integers s_i, t_i, w_i ($1 \le s_i < t_i \le n, 1 \le w_i \le 10^9$) — the endpoints of the passage and the minimum number of batteries required to pass through it.

It is guaranteed that the sum of n does not exceed $2 \cdot 10^5$.

It is guaranteed that the sum of m does not exceed $3 \cdot 10^5$.

Output

For each test case, output the minimum number of batteries that you can have at the end of the journey, or -1 if it is impossible to reach point n.

Example

input

```
4
3 3
2 0 0
1 2 1
2 3 1
1 3 2
5 6
2 2 5 0 1
1 2 2
1 3 1
1 4 3
3 5 5
2 4 4
4 5 3
2 0
1 1
4 4
3 10 0 0
1 2 1
1 3 3
2 3 10
```

```
3 4 5
```
Copy

output Copy

```
1
4
-1
10
```

Note

In the first test case, you need to take 1 battery at the starting point, then move to point 2, and then to point 3.

In the second test case, you need to take 2 batteries at the starting point, then move to point 2, take another 2 batteries, move to point 4, and then to point 5.

In the third test case, there is no path from point 1 to point n.

In the fourth test case, you need to take 1 battery at the starting point, then move to point 2, take another 9 batteries, move to point 3, and then to point 4.