# Problem E. Meteorite

Time limit 1000 ms
Memory limit 256MB

## Problem Description

A city is divided into $n$ consecutive blocks. Astronomers have discovered that a meteor shower is about to strike. To protect the city, a defense shield system can be activated.

Each block has a different importance and requires a different level of protection. Defense shields can be built, but each shield has its own restrictions.

The city consists of blocks numbered from 1 to $n$.

- Block $i$ requires at least $a_i$ units of defense strength.
- There are $m$ candidate shields. The $j$-th shield can cover a segment $[l_j, r_j]$ and provides an additional $w_j$ units of defense strength to every block in that segment.
- However, to build the $j$-th shield, the system must reach **level** at least $k_j$.

If a block is covered by multiple shields, their strengths are **added together**.

Find the minimum level the system needs to reach so that **all blocks** receive enough defense strength. If it is impossible, output $-1$.

## Input format

The first line contains an integer $t$ ($1 \le t \le 10$) — the number of test cases.

Each test case is described as follows:

The first line contains two integers $n, m$ ($1 \le n, m \le 3 \times 10^5$).

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^{12}$).

Each of the next $m$ lines contains four integers $l_j, r_j, w_j, k_j$ ($1 \le l_j \le r_j \le n$, $1 \le w_j, k_j \le 10^{12}$).

## Output format

Output a single integer — the minimum system level required to satisfy all demands, or $-1$ if it is impossible.

## Subtask score

| Subtask | Score | Additional Constraints |
|---|---|---|
| 1 | 11 | $n, m \le 1000$, $a_i, w_j, k_j \le 1000$ |
| 2 | 13 | All shield intervals are pairwise disjoint |
| 3 | 13 | All shields require the same level $k_j$ |
| 4 | 63 | No additional constraints |

# Sample

**Sample Input 1**

```
1
5 3
2 1 3 0 1
1 3 2 4
3 5 1 2
2 2 1 1
```

**Sample Output 1**

```
4
```

**Sample Input 2**

```
1
4 2
1 2 3 4
1 2 1 1
3 4 2 1
```

**Sample Output 2**

```
-1
```