Exercise – *Carsharing*

Carsharing is very popular in *Algoland*. The *Algolandians* love that they can just pick up a car and drive it from one city to another without having to return it – something that many car sharing systems in other countries do not allow.

This flexible return-policy complicates the management of the car fleet enormously though. For example, if the route from A to B is more popular than the reverse route from B to A, we might have to turn some customers down who want to drive from A to B, so that we do not run out of cars at station A.

You were hired to improve the online booking system that decides which customer can get a car and who has to take the bus. You are given the list of booking requests that contains for each customer when and where he wants to drive and how much money you would earn. You also know how the cars are distributed initially and you can assume that all rental stations have sufficient parking space and that the handover of cars happens instantaneously. You should select a set of booking request that is feasible (all customers that you select can drive their desired route) and optimal (maximizes the profit).

Input The first line of the input contains the number of test cases T, each described as follows.

- It starts with a line that contains two integers N and S, separated by a space. N denotes the number of booking requests and S the number of rental stations in *Algoland*. We have $1 \leqslant N \leqslant 10'000$ and $2 \leqslant S \leqslant 10$.
- The second line contains S space-separated integers. These numbers $l_1, \ldots, l_S$ denote the number of cars placed initially at every rental station. We have $0 \leqslant l_i \leqslant 100$ for all i.
- The remaining N lines of each test case describe the booking requests, one request per line. The i-th of these lines contains the i-th request represented as space-separated numbers s_i, t_i, d_i, a_i, p_i. These represent the indices of the start and target rental station s_i and t_i, the departure and arrival times d_i and a_i in minutes, and the profit p_i that you would make by satisfying this request. We have $1 \leqslant s_i, t_i \leqslant S$, $0 \leqslant d_i < a_i \leqslant 100'000$ and $1 \leqslant p_i \leqslant 100$.

Output For each test case output a line with a single integer p, the maximum profit that the carsharing system in *Algoland* can achieve.

Points There are five groups of test cases, worth 20 points each.

- In the first three groups of test cases, all times are at most $10'000$ and multiples of 30 minutes.
- In the first four groups of test cases, there are only two rental stations ($S = 2$).

1. For the first group of test cases, you may assume that there is only a single car ($l_1 + l_2 = 1$).

the hard part of this problem is that, if we want to use a spacial-time graph, we have to handle a huge graph with large n, this leave us only to use successive shortest path algorithm, which gives O(|f|(m+nlogn)) complexity.

Souce + sink + node at each time point = O(2N) node

each request generate 2 edges, and total |f| is at most 100, therefore, the complexity is about 100x(10000 + 10000x lg 10000)

another tricky part is how we design efficient id system to identify nodes, we have to use map

2. For the second group of test cases, you may assume that there are only at most 20 booking requests ($N \leqslant 20$).
3. For the third group of test cases, there can be many booking requests.
4. For the fourth group of test cases, the booking times are no longer constrained to half hours below $10'000$.
5. For the fifth group of test cases, there are no additional assumptions.

Corresponding sample test cases are contained in `testi.in/out`, for $i \in \{1, 2, 3, 4, 5\}$.

Sample Input

```
2
3 2
1 0
2 1 60 90 7
1 2 0 90 10
1 2 30 60 5
6 2
1 2
1 2 0 30 1
2 1 0 60 1
1 2 0 60 1
1 2 60 90 1
2 1 0 30 1
2 1 30 90 1
```

Sample Output

```
12
5
```

This task is based on the paper "*Scheduling Transfers of Resources over Time: Towards Car-Sharing with Flexible Drop-Offs*" by Kateřina Böhmová, Yann Disser, Matúš Mihalák, and Rastislav Šrámek.