Train

In the year 2992, most jobs have been taken by robots. Many people hence have abundant free time, and so is your family, who just decided to go for interstellar travel!

There are N reachable planets indexed from 0 to $N - 1$ and M interstellar train routes. The train route i ($0 \leq i < M$) starts from Planet $X[i]$ at time $A[i]$, arrives in Planet $Y[i]$ at time $B[i]$, and costs $C[i]$. Trains are the only transportation between planets, so you can only get off a train on its destination planet, and must take the next train on the same planet (transfers take no time). Formally, a sequence of trains $q[0], q[1], ..., q[P]$ is valid to be taken if and only if for any $1 \leq k \leq P$, $Y[q[k-1]] = X[q[k]]$ and $B[q[k-1]] \leq A[q[k]]$.

As interstellar travel is time-consuming, you realize that in addition to the train fare, the cost of meals is significant. Thankfully, **interstellar trains provide unlimited food for free**. That is, if you decide to take train route i, then at any time between $A[i]$ and $B[i]$ **(inclusive)** you can take any number of meals with no cost. But while your family is waiting for the next train on any Planet i, you have to pay for each meal at the cost $T[i]$.

Your family need to have W meals, and the i-th ($0 \leq i < W$) meal can be taken **instantenously** at any time between $L[i]$ and $R[i]$ **(inclusive)**.

Now at time 0, your family are on Planet 0. You need to figure out the minimum cost to reach Planet $N - 1$. If you cannot reach there, your answer should be -1.

Implementation Details

You need to implement the following function:

```
long long solve(int N, int M, int W, std::vector<int> T,
                std::vector<int> X, std::vector<int> Y,
                std::vector<int> A, std::vector<int> B, std::vector<int> C,
                std::vector<int> L, std::vector<int> R);
```

- N: The number of planets.
- M: The number of interstellar train routes.
- W: The number of meals.
- T: An array with a length of N. $T[i]$ represents the cost of each meal on Planet i.

- X, Y, A, B, C: five arrays with a length of M. The tuple $(X[i], Y[i], A[i], B[i], C[i])$ describes the i-th train route.
- L, R: Two arrays with a length of W. The pair $(L[i], R[i])$ describes the time interval to have the i-th meal.
- This function should return the minimum cost to reach Planet $N - 1$ from Planet 0 if you can reach Planet $N - 1$, and -1 if you cannot do so.
- For each test case, this function will be called exactly once.

Examples

Example 1

Consider the following call:

```
solve(3, 3, 1, {20, 30, 40}, {0, 1, 0}, {1, 2, 2},
        {1, 20, 18}, {15, 30, 40}, {10, 5, 40}, {16}, {19});
```

One way to reach planet $N - 1$ is by taking Train 0 and then Train 1, which costs 45 (the detailed calculation is shown below).

Time	Action	Cost (if any)
1	Take Train 0 in Planet 0	10
15	Arrive in Planet 1	
16	Take meal 0 in Planet 1	30
20	Take Train 1 in Planet 1	5
30	Arrive in Planet 2	

A better way to reach planet $N - 1$ is by taking Train 2 only, which costs 40 (the detailed calculation is shown below).

Time	Action	Cost (if any)
18	Take Train 2 in Planet 0	40
19	Take meal 0 on Train 2	
40	Arrive in Planet 2	

In this way of reaching planet $N - 1$, it's also valid to take meal 0 at time 18.

Therefore, the function should return 40.

Example 2

Consider the following call:

```
solve(3, 5, 6, {30, 38, 33}, {0, 1, 0, 0, 1}, {2, 0, 1, 2, 2},
        {12, 48, 26, 6, 49}, {16, 50, 28, 7, 54}, {38, 6, 23, 94, 50},
        {32, 14, 42, 37, 2, 4}, {36, 14, 45, 40, 5, 5});
```

The optimal path is to take Train 0 with a cost of 38. Meal 1 can be taken for free on Train 0. Meals 0 , 2, and 3 must be taken on Planet 2 for $33 \times 3 = 99$. Meals 4 and 5 must be taken on Planet 0 for $30 \times 2 = 60$. The total cost is $38 + 99 + 60 = 197$.

Therefore, the function should return 197.

Constraints

- $2 \le N \le 10^5$.
- $0 \le M, W \le 10^5$.
- $0 \le X[i], Y[i] < N, X[i] \ne Y[i]$.
- $1 \le A[i] < B[i] \le 10^9$.
- $1 \le T[i], C[i] \le 10^9$.
- $1 \le L[i] \le R[i] \le 10^9$.

Subtasks

1. (5 points): $N, M, A[i], B[i], L[i], R[i] \le 10^3$ and $W \le 10$.
2. (5 points): $W = 0$.
3. (30 points): No two meals overlap in time. Formally, for any time z where $1 \le z \le 10^9$, there is at most one i ($0 \le i < W$) such that $L[i] \le z \le R[i]$.
4. (60 points): No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

- Line 1: $N\ M\ W$
- Line 2: $T[0]\ T[1]\ T[2]\ \cdots\ T[N-1]$
- Line $3 + i$ ($0 \le i < M$): $X[i]\ Y[i]\ A[i]\ B[i]\ C[i]$
- Line $3 + M + i$ ($0 \le i < W$): $L[i]\ R[i]$

The sample grader prints your answers in the following format:

- Line 1: the return value of solve