

은하철도

2992년 대부분의 일은 로봇이 하고 있다. 사람들은 자유시간이 많고, 여러분 가족도 마찬가지로 은하철도를 이용하여 우주 여행을 하기로 했다.

0부터 $N - 1$ 까지 정수로 번호가 매겨진 N 개의 행성을 여행할 수 있고, 은하철도에는 M 대의 열차가 운행한다. 열차 i ($0 \leq i < M$)는 행성 $X[i]$ 를 $A[i]$ 시각에 출발해서 행성 $Y[i]$ 에 $B[i]$ 시각에 도착하며, 표값이 $C[i]$ 이다. 행성간을 이동하는 방법은 열차밖에 없기 때문에, 한 행성에서 내린 다음 이 행성에서 출발하는 열차를 타는 식으로만 이동할 수 있다. (열차를 갈아타는데는 시간이 걸리지 않는다) 열차의 순열 $q[0], q[1], \dots, q[P]$ 이 타당하다는 것은 어떤 $1 \leq k \leq P$ 에 대해서도, $Y[q[k - 1]] = X[q[k]]$ 이고 $B[q[k - 1]] \leq A[q[k]]$ 라는 것과 같은 뜻이다.

우주 여행은 시간이 많이 걸리기 때문에, 표값만 돈이 나가는게 아니고 밥값도 많이 든다는 것을 알게 되었다. 다행히도, **열차를 타고 있는 동안은 음식이 무료로 제공된다.** 그러나 행성 i 에서 열차를 기다리는 동안에는, 매끼마다 밥값이 $T[i]$ 가 든다.

여러분의 가족은 W 번 식사를 해야 하고, i 번째 식사는 ($0 \leq i < W$) $L[i]$ 시각 이후, $R[i]$ 시각 이전에 (**경계 포함**) 먹어야 한다. 식사를 하는데는 **시간이 걸리지 않는다.**

시각 0에 여러분 가족은 행성 0에 있다. 행성 $N - 1$ 까지 도착하는데 걸리는 최소의 비용을 구하시오. 만약 이 행성에 도착할 수 없다면, 여러분의 답은 -1 이어야 한다.

Implementation Details

다음 함수를 구현해야 한다.

```
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 : 행성의 수
- M : 은하철도를 달리는 열차의 수
- W : 식사 횟수
- T : 길이 N 인 배열. $T[i]$ 는 행성 i 에서 한 끼 식사 비용.
- X, Y, A, B, C : 길이 M 인 다섯 개의 배열. 순서쌍 $(X[i], Y[i], A[i], B[i], C[i])$ 는 열차 i 에 대한 정보이다.
- L, R : 길이 W 인 두 개의 배열. 순서쌍 $(L[i], R[i])$ 은 i 번째 식사를 할 수 있는 기간을 나타낸다.

- 이 함수의 리턴값은 행성 0에서 행성 $N - 1$ 까지 여행하는데 드는 최소 비용이다. 만약 행성 $N - 1$ 에 도착할 수 없다면 -1 을 리턴한다.
- 각각의 테스트 케이스에 대해서 이 함수는 정확히 한 번 호출된다.

Examples

Example 1

다음 호출을 생각해보자.

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

행성 2에 도착하는 한 가지 방법은 열차 0을 탄 다음 열차 1로 갈아타는 것으로, 비용이 45가 든다. (자세한 계산은 다음 표를 참조)

Time	Action	Cost (if any)
1	행성 0에서 열차 0을 탄다	10
15	행성 1 도착	
16	행성 1에서 식사 0	30
20	행성 1에서 열차 1을 탄다	5
30	행성 2 도착	

행성 2에 도착하는 더 좋은 방법은 열차 2만 타고 가는 방법으로 비용이 40이 든다. (자세한 계산은 다음 표를 참조)

Time	Action	Cost (if any)
18	행성 0에서 열차 2를 탄다	40
19	열차 2에서 식사 0	
40	행성 2 도착	

따라서, 이 함수의 리턴값은 40이어야 한다.

Example 2

다음 호출을 생각해보자.

```
solve(3, 10, 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});
```

최적의 경로는 차비 38로 열차 0을 타는 것이다. 식사 1은 열차 0에서 무료로 먹을 수 있다. 식사 0, 2, 3은 행성 2에서 먹는데 비용은 $33 \times 3 = 99$ 이다. 식사 4와 5는 행성 0에서 먹는데 비용은 $30 \times 2 = 60$ 이다. 총 비용은 $38 + 99 + 60 = 197$.

따라서, 이 함수의 리턴값은 197이어야 한다.

Constraints

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

Subtasks

1. (5 points): $N, M, A[i], B[i], L[i], R[i] \leq 10^3$ and $W \leq 10$.
2. (5 points): $W = 0$.
3. (30 points): 어떤 두 식사도 먹을 수 있는 시간이 서로 교차하지 않는다. 엄밀하게는, $1 \leq z \leq 10^9$ 인 어떤 시간 z 에 대해서도, $L[i] \leq z \leq R[i]$ 인 i 는 최대 하나이다 ($0 \leq i < W$).
4. (60 points): 추가적인 제약 조건이 없다.

Sample Grader

샘플 그레이더는 다음 양식으로 입력을 읽는다.

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

샘플 그레이더는 여러분의 답을 다음 양식으로 출력한다.

- Line 1: solve의 리턴값