

Train

В 2992 году большинство рабочих мест заняли роботы. Поэтому у многих людей появилось много свободного времени, как и у вашей семьи, которая только что решила отправиться в межзвездное путешествие!

Существует N достижимых планет, пронумерованных от 0 до $N - 1$, и M маршрутов межзвездных поездов. Маршрут поезда i ($0 \leq i < M$) начинается с планеты $X[i]$ в момент времени $A[i]$, прибывает на планету $Y[i]$ в момент времени $B[i]$ и стоит $C[i]$. Поезда - единственный вид транспорта между планетами, поэтому вы можете сойти с поезда только на планете назначения и должны сесть на следующий поезд на той же планете (пересадки не занимают времени). Формально, последовательность поездов $q[0], q[1], \dots, q[P]$ является допустимой тогда и только тогда, когда для любого $1 \leq k \leq P$, $Y[q[k - 1]] = X[q[k]]$ и $B[q[k - 1]] \leq A[q[k]]$.

Поскольку межзвездные путешествия отнимают много времени, вы понимаете, что помимо платы за проезд в поезде, расходы на питание весьма значительны. К счастью, **в межзвездных поездах еда предоставляется бесплатно в неограниченном количестве**. То есть, если вы решили ехать поездом по маршруту i , то в любое время между $A[i]$ и $B[i]$ (**включительно**) вы можете взять любое количество еды без каких-либо затрат. Но пока ваша семья ждет следующего поезда на любой планете i , вам придется платить за каждый прием пищи по цене $T[i]$.

Вашей семье необходимо W еды, и i -ю ($0 \leq i < W$) еду можно взять **моментально** в любое время между $L[i]$ и $R[i]$ (**включительно**).

Сейчас, в момент времени 0, ваша семья находится на планете 0. Вам нужно определить минимальную стоимость достижения планеты $N - 1$. Если вы не можете туда добраться, ваш ответ должен быть -1 .

Implementation Details

Вам необходимо реализовать следующую функцию:

```
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 : Количество планет.
- M : Количество маршрутов межзвездных поездов.
- W : Количество приемов пищи.
- T : Массив длиной N . $T[i]$ представляет собой стоимость каждого приема пищи на планете i .
- X, Y, A, B, C : пять массивов длиной M . Набор $(X[i], Y[i], A[i], B[i], C[i])$ описывает маршрут i -го поезда.
- L, R : Два массива длиной W . Пара $(L[i], R[i])$ описывает временной интервал для i -го приема пищи.
- Эта функция должна возвращать минимальную стоимость достижения планеты $N - 1$ с планеты 0, если вы можете достичь планеты $N - 1$, и -1 , если вы не можете этого сделать.
- Для каждого тестового случая эта функция будет вызвана ровно один раз.

Examples

Example 1

Рассмотрим следующий вызов:

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

Один из способов добраться до планеты $N - 1$ - это сесть на поезд 0, а затем на поезд 1, что обойдется в 45 (подробный подсчёт показан ниже).

Время	Действие	Стоимость (если есть)
1	Сесть на поезд 0 на планете 0	10
15	Прибытие на планету 1	
16	Взять еду 0 в Планете 1	30
20	Сесть на поезд 1 в Планете 1	5
30	Прибыть на планету 2	

Лучшим способом добраться до планеты N является поездка только на поезде 2, которая стоит 40 (подробный подсчёт приведён ниже).

Время	Действие	Стоимость (если есть)
18	Сесть на поезд 2 в Планете 0	40
19	Перекусить 0 в поезде 2	
40	Прибыть на планету 2	

При таком способе достижения планеты $N - 1$ также допустимо взять еду 0 в момент времени 18.

Таким образом, функция должна вернуть 40.

Example 2

Рассмотрим следующий вызов:

```
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});
```

Оптимальный путь - это поездка на поезде 0 с затратами 38. Еду 1 можно взять бесплатно в поезде 0. Еда 0, 2 и 3 должна быть взята на планете 2 за $33 \times 3 = 99$. Еда 4 и 5 должна быть взята на планете 0 за $30 \times 2 = 60$. Общая стоимость составляет $38 + 99 + 60 = 197$.

Поэтому функция должна вернуть 197.

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 баллов): $N, M, A[i], B[i], L[i], R[i] \leq 10^3$ и $W \leq 10$.
2. (5 баллов): $W = 0$.
3. (30 баллов): Никакие два приема пищи не пересекаются по времени. Формально, для любого времени z , где $1 \leq z \leq 10^9$, существует не более одного i ($0 \leq i < W$) такого, что $L[i] \leq z \leq R[i]$.
4. (60 баллов): Никаких дополнительных ограничений.

Sample Grader

Образец грейдера считывает входные данные в следующем формате:

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

Образец грейдера выводит ваши ответы в следующем формате:

- Строка 1: возвращаемое значение функции `solve`.