Pan African Olympiad in Informatics Team Selection Test
2025

Water Cooling

Time limit: 2 seconds **Memory limit: 512 MB**

Ethan is working on programming a cooling system for his giant robot tower. It is 100 meters tall, separated into N sections: the ith ($0 \leq i < N$) section covers an inclusive range $(S[i], T[i])$ $(1 \leq S[i] < T[i] \leq 100)$ of the tower's height, given in meters, and needs to be cooled down by $C[i]$ units. All sections are stacked on top of each other and don't overlap.

Ethan has a large water tube going along the tower and M $(1 \leq M \leq 10)$ cooling pumps: the jth $(0 \leq j < M)$ pump supplies the inclusive height range of $(A[j], B[j])$ $(1 \leq A[j] < B[j] \leq 100)$ meters with cold water, cooling that range by $P[j]$ units at a cost of $D[j]$. Sometimes, however, these ranges aren't disjoint as the plumbing of two sections gets intertwined to make changing lubricant and hot water easier.

Since Ethan can't be good at both mechanical engineering and programming, he has tasked you with writing a simple and efficient algorithm to figure out the cost of the cheapest configuration(s) of his system that fully cools down all sections of the tower.

Problem Description

- You are given arrays S, T and C of length N: For each i $(0 \leq i < N)$, the inclusive range $(S[i], T[i])$ $(1 \leq S[i] < T[i] \leq 100)$ describes a subarray of the inclusive range $(1, 100)$ which needs a cooling of $C[i]$ units, such that all ranges described by S and T are disjoint.
- You are also given arrays A, B, P and D of length M: For each j $(0 \leq j < M)$, you can cool the inclusive range $(A[j], B[j])$ $(1 \leq A[j] < B[j] \leq 100)$ by $P[j]$ units at a cost of $D[j]$; note that not all ranges described by A and B are necessarily disjoint, and selecting two overlapping ranges $R_i = (A[i], B[i])$ and $R_j = (A[j], B[j])$ $(0 \leq i, j < M)$ cools down their intersection $R_i \cap R_j$ by $P[i] + P[j]$ units.

Find the minimum cost (denoted as C_{min}) for which the tower is properly cooled, as in the amount of provided cooling is superior or equal to the amount of needed cooling for all ranges described by S and T.

Input

Input is formatted as follows:

```
N M
S[0] T[0] C[0]
S[1] T[1] C[1]
...
S[N-1] T[N-1] C[N-1]
A[0] B[0] P[0] D[0]
A[1] B[1] P[1] D[1]
...
A[M-1] B[M-1] P[M-1] D[M-1]
```

Output

Output is expected as follows:

```
Cmin
```

Constraints

- $1 \leq N \leq 20$
- $1 \leq M \leq 10$
- $1 \leq S[i] < T[i] \leq 100 \ (0 \leq i < N)$
- $1 \leq A[j] < B[j] \leq 100 \ (0 \leq j < M)$
- $1 \leq C[i] \leq 10^6 \ (0 \leq i < N)$
- $1 \leq P[j] \leq 10^6 \ (0 \leq j < M)$

Subtasks

For this task, the amount of points you are awarded is the maximum of the percentages of testcases correctly answered across all of your submissions. This means that if this task has t testcases and you have answered at most s $(s \leq t)$ testcases correctly across all of your submissions, the amount of points you are awarded will be $100 * \frac{s}{t}$.

Example

```
2 4
1 5 2
7 9 3
2 9 2 3
1 6 2 8
1 2 4 2
6 9 1 5
```

Output:

```
10
```

Explanation

One possible solution that results in the least amount of money spent is to select those that cool the ranges (2,9), (1,2), and (6,9), for a cost of 3+2+5=10. One can verify that all ranges have been properly cooled by adding up the amounts that each range cools up: for example, we can check that at a height of 2 meters, the system provides a cooling of $4 + 2 = 6$, which is greater than or equal to the amount needed ($6 \geq 2$). The program should then output 10.