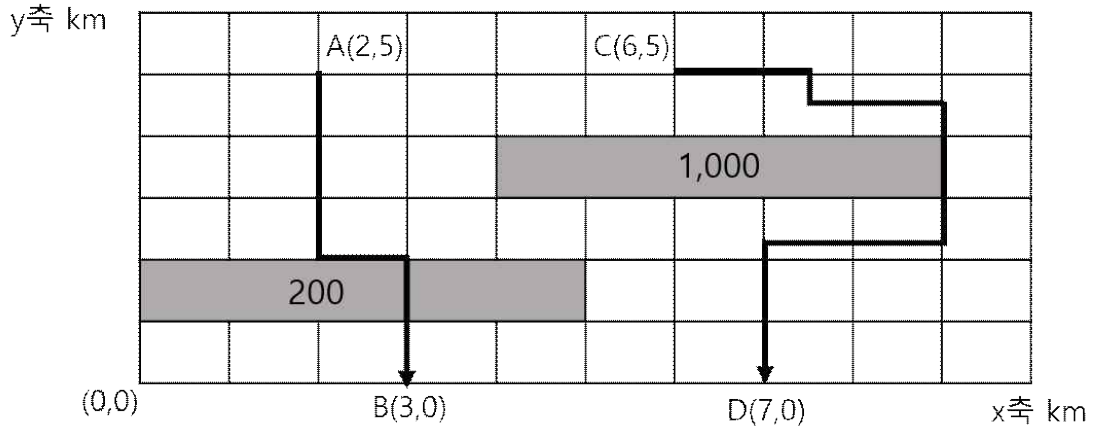
문제4 : 대평원

1km를 100분에 이동하는 개미가 대평원 위의 시작점에서 도착점으로 이동하려고 한다. 단, x축과 y축에 평행하게 이동해야 하고 구릉지에서는 이동 시간이 늘어날 수 있다. 여러분은 시작점, 도착점, 구릉지들, 그리고 구릉지마다 1km를 이동하는 데 걸리는 시간이 주어졌을 때 이 개미가 시작점에서 도착점까지의 가장 빠르게 이동할 수 있는 시간을 계산하라.

구릉지는 x축과 y축에 평행한 변으로 구성된 직사각형 모양이며 직사각형마다 1km를 이동하는 데 걸리는 시간이 주어진다. 이 이동시간은 직사각형의 내부에만 적용되고 변에는 적용되지 않는다. 또한 서로 다른 두 직사각형은 겹치지 않으며 시작점, 도착점, 그리고 서로 다른 직사각형들의 꼭지점 x좌표들은 모두 정수이며 겹치지 않는다. y좌표들도 마찬가지로 모두 정수이며 겹치지 않는다. 시작점과 도착점은 항상 직사각형의 외부에 존재한다.



위 예제는 km 단위 좌표에 km당 이동 시간이 200분인 직사각형-왼쪽 아래 좌표가 (0,1)이고 오른쪽 위 좌표가 (5,2)-과 km당 이동 시간이 1,000분인 직사각형-왼쪽 아래 좌표가 (4,3)이고 오른쪽 위 좌표가 (9,4)-를 표시한 것이다. 시작점이 A이고 도착점이 B인 경우와 시작점이 C이고 도착점이 D인 경우의 최단시간 경로가 표시되어 있으며 각각 최단시간은 700분과 1,000분이다.

여러분은 다음 함수를 구현해야만 한다.

- long long shortest_path(pair<int, int> src, pair<int, int> dst, vector<pair<int, int>> p1, vector<pair<int, int>> p2, vector<int> w); 단 한 번 호출되는 함수이다. src와 dst는 시작점과 도착점이다. 각 직사각형의 왼쪽 아래점은 p1에 오른쪽 위의 점은 p2에 주어진다. 주어진 값을 이용하여 src로부터 dst까지의 최단시간을 구하여 return 한다.

구현 세부사항

여러분은 plain.cpp라는 이름을 가진 하나의 파일을 제출해야만 한다. 이 파일에는 다음의 함수가 구현되어 있어야 한다.

- long long shortest_path(pair<int, int> src, pair<int, int> dst, vector<pair<int, int>> p1, vector<pair<int, int>> p2, vector<int> w);

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론, 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 읽는다. x, y 좌표 값의 단위는 km이다.

- line 1: N x_s y_s x_e y_e
  - N: 직사각형의 개수,
  - x_s, y_s: 시작점의 x, y좌표,
  - x_e, y_e: 도착점의 x, y좌표

- 다음 N개의 줄 각각: x_1 y_1 x_2 y_2 w
  - x_1 y_1: 직사각형의 왼쪽 아래 꼭지점의 x, y 좌표,
  - x_2 y_2: 직사각형의 오른쪽 위 꼭지점의 x, y 좌표,
  - $x_1 < x_2$,
  - $y_1 < y_2$,
  - w: 직사각형 내부에서 1km 이동 시간(분)

주어진 grader는 여러분의 코드가 shortest_path() 함수에서 리턴한 값을 출력한다.

제한 조건

- $1 \leq N \leq 100{,}000$
- $100 \leq w \leq 10^8$
- $0 \leq x, y \leq 10^8$ (x, y는 시작점, 도착점, 또는 직사각형의 꼭지점의 좌표값)
- N, w, x, y는 모두 정수

서브태스크 1 [23 points]

- $N \leq 500$

서브태스크 2 [35 points]

- $N \leq 5{,}000$

서브태스크 3 [31 points]

- $x_2 - x_1 = 1$

서브태스크 4 [61 points]

- 추가 제한이 없다.

[입력 예 1]

```
3 2 14 5 1
4 6 6 10 1000
0 7 3 9 200
1 2 8 5 150
```

[출력 예 1]

```
1750
```

[입력 예 2]

```
13 0 38 100 25
1 39 2 46 190
9 78 10 80 230
20 42 21 89 170
27 26 28 68 170
35 41 36 99 270
43 36 44 63 280
51 15 52 27 150
57 14 58 29 190
64 2 65 90 160
75 33 76 35 290
78 5 79 100 290
88 28 89 40 190
94 7 95 50 250
```

[출력 예 2]

```
11770
```