J - Prostokąty

Opis

Mamy płaszczyznę podzieloną na kwadraciki 1x1. Kwadraciki mają boki równoległe do osi układu współrzędnych, a ich wierzchołki mają współrzędne całkowite. Kwadraciki nie mają ze sobą wspólnego wnętrza (nachodzą na siebie jedynie bokami i wierzchołkami) i nie ma żadnej pustej przestrzeni między nimi (tj. pokrywają dokładnie całą płaszczyznę).

Początkowo wszystkie kwadraciki są koloru białego. Na płaszczyznę kładziemy serię prostokątów o wierzchołkach z całkowitymi współrzędnymi i bokach równoległych do osi. Położenie jednego takiego prostokąta w praktyce oznacza negację koloru wszystkich kwadracików w nim zawartych. Przez negację rozumiemy zamianę koloru na czarny jeśli aktualny kolor jest biały, albo na biały jeśli kolor jest czarny.

Zadanie

Dla podanej listy prostokątów Twój program powinien wyliczyć ile kwadracików będzie miało kolor czarny po położeniu wszystkich tych prostokątów na płaszczyznę.

Specyfikacja wejścia

W pierwszym wierszu wejścia znajduje się jedna dodatnia liczba całkowita, oznaczająca liczbę zestawów testowych, które dalej pojawią się na wejściu. Każdy zestaw ma następującą postać. W pierwszym wierszu znajduje się jedna liczba całkowita N ($1 \leq N \leq 100.000$), oznaczająca liczbę prostokątów kładzionych na płaszczyznę. W kolejnych N wierszach znajdują się czwórki liczb całkowitych $x1$, $y1$, $x2$, $y2$, oddzielonych pojedynczą spacją ($-1.000.000.000 \leq x1,y1,x2,y2 \leq 1.000.000.000$, $x1 \neq x2$, $y1 \neq y2$). Pary ($x1$,$y1$) i ($x2$,$y2$) to współrzędne przeciwległych wierzchołków prostokątów.

Specyfikacja wyjścia

Dla każdego zestawu danych pojawiającego się na wejściu należy wypisać dokładnie jedną liczbę całkowitą (każdą w osobnej linii), oznaczającą liczbę czarnych kwadracików powstałych po położeniu wszystkich prostokątów opisanych w zestawie. [Uwaga! Liczba ta może nie mieścić się w 32-bitowym typie całkowitym. Zalecamy użycie typu *long long* w C/C++ oraz *qword* w Pascalu - są to typy 64-bitowe.]

Przykład

Wejście

```
3
2
-1 -2 1 2
```

```
-2 -1 2 1
3
10 10 0 0
0 10 5 0
10 10 5 0
1
0 0 1000000 1000000
```

Wyjście

```
8
0
1000000000000
```