

Task 2. Delivery

Мэттью - владелец транспортной компании. Он живет в городе, в котором на одной прямой расположены ровно 10^9 домов. Каждый дом имеет номер, дом с номером i имеет соседей с номерами $i - 1$ и $i + 1$ (если они существуют). Компания Мэттью получила N запросов вида выполнить доставку в дом H_i во время T_i . Нет двух запросов о доставке в один и тот же дом в одно и то же время. Для экономии денег Мэттью хочет узнать, какое минимальное число грузовиков потребуется для выполнения всех запросов. Каждый грузовик, который может купить Мэттью, перемещается ровно на 1 дом налево или направо за одну единицу времени (они также могут не перемещаться и остаться у того же дома). Изначально все грузовики могут быть расположены где угодно по выбору Мэттью. Временем самой доставки можно пренебречь.

Мэттью - очень занятой человек, у него нет времени на простейшие задачи вроде этой, поэтому он просит вас написать программу `delivery.cpp`, которая вычисляет минимальное число грузовиков.

Input

В первой строке входных данных содержится единственное число N – количество запросов. Каждая из следующих N строк содержит по два числа T_i и H_i – время и место доставки

Output

В единственной строке выведите одно число - искомое минимальное число грузовиков.

Constraints

$$1 \leq N \leq 10^6$$

$$1 \leq T_i, H_i \leq 10^9$$

$$T_i \neq T_j \text{ or } H_i \neq H_j \text{ for } i \neq j$$

Subtasks

Подзадача	Баллы	N
1	25	$\leq 10^3$
2	10	$\leq 10^4$
3	40	$\leq 2 \times 10^5$
4	20	$\leq 10^6$

Example 1

Input	Output	Explanation
6 1 1 2 3 3 2 5 4 4 1 4 3	2	<p>Минимальное число грузовиков - 2.</p> <p>Один из способов выполнить все запросы:</p> <p>Первый грузовик: $(1, 1)^* \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (4, 1)^* \rightarrow (5, 1)$</p> <p>Второй грузовик: $(1, 2) \rightarrow (2, 3)^* \rightarrow (3, 2)^* \rightarrow (4, 3)^* \rightarrow (5, 4)^*$</p> <p>Здесь (t, h) обозначает время нахождения грузовика у дома h во время t, а символом $*$ отмечены те моменты, в которые происходит доставка.</p>