



Muchas Tareas

Berta tiene una lista de n tareas. La i -ésima tarea requiere t_i segundos para ser completada. Además, cada tarea tiene un *plazo* p_i : la i -ésima tarea debe ser completada antes de que pasen más de $p_i + t_i$ segundos desde el inicio (por tanto, debe ser iniciada en el segundo p_i o antes).

Berta quiere realizar las tareas estrictamente en el orden en que están escritas en su lista, pudiendo omitir algunas de ellas.

Ayuda a Berta a determinar cuántas tareas de su lista puede completar como máximo.

Entrada y salida

La primera línea contiene un entero T , el número de casos a procesar.

Cada caso comienza con una línea con un entero n , el número de tareas. A continuación, siguen n líneas, la i -ésima de ellas con los dos enteros t_i y p_i .

Por cada caso, imprime una línea con el máximo número de tareas que se pueden completar.

Ejemplo

Entrada:

```
3
5
2 0
1 1
1 1
2 2
3 4
2
10000 100000000
1 9999
10
8 25
5 12
7 10
11 28
3 18
6 32
10 45
5 34
7 28
6 42
```

Salida:

```
4
1
7
```



Restricciones

- $1 \leq T \leq 10^3$.
- $1 \leq n \leq 10^3$. La suma de n^2 sobre todos los casos es como mucho 10^6 .
- $1 \leq t_i \leq 10^6$ para todo $i = 1, \dots, n$.
- $0 \leq p_i \leq 10^9$ para todo $i = 1, \dots, n$.

Subtareas

1. (39 puntos) $t_i \leq n$, $p_i \leq n$ y la suma de n^3 sobre todos los casos es como mucho 10^6 .
2. (15 puntos) Todos los valores de $p_i + t_i$ son iguales: $p_1 + t_1 = p_2 + t_2 = \dots = p_n + t_n$.
3. (16 puntos) Todos los valores de p_i son iguales: $p_1 = p_2 = \dots = p_n$.
4. (30 puntos) Sin restricciones adicionales.