

Ejercicio 1. Algoritmos voraces y colas de prioridad (3.5 puntos)

En breve comenzará la *Semana de la Informática*. La facultad ha organizado multitud de actividades (conferencias, seminarios, talleres, jornadas de puertas abiertas). Yo he estado mirando la planificación y he ido apuntando todas las actividades que me parecen interesantes. El problema es que son tantas que algunas solapan en el tiempo, por lo que no podré asistir a todas. Lo que he pensado es convencer a algunos compañeros para que asistan a las actividades a las que no puedo ir yo para que entre todos asistamos a todas las actividades que me han parecido interesantes, y después compartamos lo aprendido.

Dada la lista de actividades, de las que conozco cuándo empiezan y terminan, ¿podrías ayudarme a calcular cuántos compañeros necesito como mínimo para cubrir todas las actividades? Puedes tener en cuenta que aunque ni mis compañeros ni yo tenemos el don de la ubicuidad, sí tenemos el poder de teletransportarnos, por lo que en cuanto termina una actividad podemos *saltar* a donde comience otra, sin perdernos nada.

Entrada

La entrada consta de una serie de casos de prueba. Cada uno comienza con una línea con el número N de actividades interesantes ($1 \leq N \leq 200.000$). A continuación aparecen N líneas, cada una con dos números que representan el momento de comienzo y de finalización de una actividad (el comienzo siempre es estrictamente menor que la finalización). Estos tiempos son números enteros entre 0 y 10^9 .

La entrada terminará con un caso sin actividades ($N = 0$), que no debe procesarse.

Salida

Para cada caso de prueba se escribirá una línea con el mínimo número de compañeros que harán posible que, junto conmigo, podamos asistir a todas las actividades, y de tal forma que ninguno tenga que estar a la vez asistiendo a dos de ellas.

Entrada de ejemplo

```
3
1 5
3 10
6 12
2
5 10
1 5
3
1 5
2 6
3 7
0
```

Salida de ejemplo

```
1
0
2
```