

Intervalos disjuntos

De n actividades conocemos sus instantes de comienzo c_i y finalización f_i ; es decir, el intervalo de realización de la actividad i ($0 \leq i < n$) es $[c_i, f_i]$. Diseñar un algoritmo de vuelta atrás que imprima todos los subconjuntos con r o mas actividades que no solapen entre si (es decir cuya intersección dos a dos es vacía)

Se puede suponer que los intervalos vienen dados en orden de comienzo.

Problema obtenido del examen de junio de 2015.

Requisitos de implementación

Hacer una estimación que considere si es posible completar el subconjunto de intervalos para que tenga más de r elementos.

Se recomienda implementar un programa que obtenga los intervalos además de contarlos y una vez que funcionen todos los casos de prueba, se pase por el juez dando como salida únicamente en número de intervalos.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de dos líneas, en la primera se indica el número de intervalos y el valor de r , en la segunda línea los valores de comienzo y final de cada intervalo ordenados por orden creciente de inicio de intervalo. Entre intervalos con el mismo comienzo no se garantiza ningún tipo de orden.

El número de actividades (intervalos) es siempre mayor que cero. El comienzo y el final del intervalo pueden ser iguales, pero el final nunca será menor que el comienzo.

Salida

Para cada caso de prueba se escribe en una línea el número de subconjuntos encontrados.

Si no existen ningún subconjunto con un número de elementos mayor que r se escribirá un cero.

Entrada de ejemplo

```
5 1
1 3 1 5 2 6 3 4 5 6
4 2
1 4 3 6 4 8 7 9
```

Salida de ejemplo

```
7
2
```

Casos de ejemplo con los intervalos calculados para realizar pruebas:

Entrada de ejemplo

```
5 1
1 3 1 5 2 6 3 4 5 6
4 2
1 4 3 6 4 8 7 9
```

Salida de ejemplo

```
7
[1,3]
[1,3] [5,6]
[1,5]
[2,6]
[3,4]
[3,4] [5,6]
[5,6]
2
[1,4] [7,9]
[3,6] [7,9]
```

Autor: Isabel Pita.