

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

Se recomienda implementar el programa sin ordenar los subconjuntos en la salida y una vez que funcionen todos los casos de prueba, se proceda a ordenarlos para pasar el programa por el juez.

Para que los subconjuntos se ordenen, se recomienda almacenar todos los subconjuntos en una matriz. A continuación ordenar la matriz utilizando la función *sort* con un método de comparación de intervalos.

```
struct intervalo {
    int inicio, fin;
};

std::istream& operator>> (std::istream& in, intervalo& p) {
    in >> p.inicio >> p.fin;
    return in;
}

bool operator==(intervalo const& p1, intervalo const& p2) {
    return p1.inicio == p2.inicio && p1.fin == p2.fin;
}

class comp {
public:
    bool operator()(std::vector<intervalo> const& v1, std::vector<intervalo> const& v2) {
        int i = 0;
        while (i < v1.size() && i < v2.size() && v1[i] == v2[i]) ++i;
        if (i < v1.size() && i < v2.size() && !(v1[i] == v2[i])) {
            if (v1[i].inicio < v2[i].inicio) return true;
            else if (v1[i].inicio > v2[i].inicio) return false;
            else if (v1[i].fin < v2[i].fin) return true;
            else if (v1[i].fin >= v2[i].fin) return false;
        }
        if (i < v1.size() || v1.size() == v2.size()) return false;
        else return true;
    }
};

std::vector<std::vector<intervalo> > todasSol;

sort(todasSol.begin(), todasSol.end(), comp());
```

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.

Salida

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

A continuación se escriben todos los subconjuntos de actividades que no solapan entre si, cada subconjunto en una línea diferente. Cada intervalo se escribe entre corchetes y los valores del intervalo se separan por una coma. Los subconjuntos se ordenan por el principio de los intervalos.

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.