



Nº matrícula: _____ Nombre: _____

Apellidos: _____

Problema. Sea U el conjunto de números enteros consecutivos en el rango $[0..n]$, denominado universo. Sea S un grupo de conjuntos cuya unión abarca todo el universo U . Se pretende encontrar el número mínimo de conjuntos de S cuya unión contenga todos los elementos de U .

Ejemplo: sea $U=\{0,1,2,3,4,5\}$. Si $S=\{\{1,2,0,3\}, \{2,4\}, \{3,4\}, \{4,5\}\}$, el número mínimo de conjuntos de S cuya unión contiene todos los elementos de U está formado por $\{\{1,2,0,3\}, \{4,5\}\}$. Si $S=\{\{1,2\}, \{2,4\}, \{0,3,4\}, \{4,5\}\}$, el número mínimo de conjuntos de S cuya unión contiene todos los elementos de U está formado por $\{\{1,2\}, \{0,3,4\}, \{4,5\}\}$.

SE PIDE: Implementar en Java el método *conjuntoMinimo*, basado en el **esquema voraz**, para que dado un universo U y un grupo de conjuntos S , devuelva **los conjuntos de S que forman el número mínimo de conjuntos** cuya unión contiene todos los elementos de U :

`ArrayList<Conjunto> conjuntoMinimo(int[] U, ArrayList<Conjunto> S)`

donde la clase *Conjunto* tiene el siguiente aspecto:

```
public class Conjunto {
    private int[] contenido;

    public Conjunto(int[] contenido){
        this.contenido = new int[contenido.length];
        for (int i=0; i<contenido.length; i++) this.contenido[i]=contenido[i];
    }
    public int longitud(){ return this.contenido.length; }
    public int get(int posicion){ return contenido[posicion]; }
}
```

```
ArrayList<Conjunto> conjuntoMinimo(int[] U, ArrayList<Conjunto> S){
    ArrayList<Conjunto> solucion = new ArrayList<Conjunto>();
    boolean[] incluidos = new boolean[U.length];
    for (int i=0; i<incluidos.length; i++) incluidos[i]=false;
    while (!todosIncluidos(incluidos)){
        Conjunto c = seleccionarCandidato(S, incluidos);
        S.remove(c);
        solucion.add(c);
        for (int i=0; i<c.longitud(); i++) incluidos[c.get(i)]=true;
    }
    return solucion;
}

boolean todosIncluidos(boolean[] incluidos){
    boolean ok=true; int i=0;
    while ((i<incluidos.length) && ok) {
        ok = incluidos[i]; i++;
    }
    return ok;
}
```

```
Conjunto seleccionarCandidato(ArrayList<Conjunto> S, boolean[] incluidos){
    Conjunto cMejor = null;
    int mejor=0, actual;
    for (Conjunto c: S){
        actual = contarDistintos(c, incluidos);
        if ( actual > mejor){
            cMejor = c;
            mejor = actual;
        }
    }
    return cMejor;
}

int contarDistintos(Conjunto c, boolean[] incluidos){
    int n=0;
    for (int i=0; i<c.longitud(); i++)
        if (!incluidos[c.get(i)]) n++;
    return n;
}
```