

Algoritmos y Estructuras de Datos.

TPL1. 1er Trabajo Práctico de Laboratorio. [2013-08-31]

Instrucciones

- El examen consiste en que escriban las funciones descriptas más abajo; implementándolas en C++ de tal forma que el código que escriban **compile y corra correctamente**, es decir, no se aceptará un código que de algún error de compilación o que tire alguna excepción/señal de interrupción en runtime. Básicamente se hace una evaluación de caja negra, aunque le daremos un rápido vistazo al código.
- Pueden utilizar todas las funciones y utilidades del estándar de C++ que por supuesto contiene a la librería STL.

1. split-mod

Escribir una función `void split_mod(list<int> &L, int m, vector<list<int> > &VL);` que dada una lista **L**, y un entero **m** divide a la lista en las sublistas de los elementos que son congruentes entre sí módulo **m**. Es decir, si **m=2** debe dejar en **VL[0]** los elementos pares y en **VL[1]** los elementos impares. Si **m=3** debe dejar en **VL[0]** los múltiplos de 3, en **VL[1]** los **x** tales que **x%3==1** y así siguiendo. En general en **VL[j]** deben quedar los elementos tales que **x%m==j**. Por ejemplo si **L=(1, 3, 2, 1, 5, 6, 0)** y **m=3**, debe quedar

```
VL[0] = (3, 6, 0)
VL[1] = (1, 1)
VL[2] = (2, 5)
```

Los elementos deben quedar en las listas **VL[j]** en el mismo orden que estaban en **L**.

2. is-sublist

Escribir una función predicado `bool is_sublist(list<int> &L1, list<int> &L2);` que determina si **L2** es una sublista de **L1** es decir si **L2** se puede obtener de **L1** sólo borrando elementos (pero sin insertar). Por ejemplo

```
L1=(1 3 2 5 6) L2=(3 5) -> true
L1=(1 3 2 5 6) L2=(3 5 2) -> false
```

Ayuda: Tomar dos posiciones **p1** y **p2**. Avanzar **p1** hasta encontrar el elemento en **p2**. Si no lo encuentra retorna falso, si lo encuentra avanza ambas posiciones.

3. max-sublist

Escribir una función `void max_sublist(list<int> &L, list<int> &subl);` que dada una lista **L** retorna la máxima sublista contigua de **L** con elementos pares **subl**. Por ejemplo si

```
L=(5, 4, 5, 8, 6, 2, 10, 9, 3, 1) -> subl=(8, 6, 2, 10)
```

Si hay varias sublistas con la misma longitud debe retornar la primera. Por ejemplo

```
L=(1, 2, 4, 6, 5, 0, 2, 4) -> subl=(2, 4, 6)
```

Ayuda: Guardar la máxima lista actual en **subl**. Recorrer la lista **L** y cada vez que se encuentra un elemento par, copiarlo junto todos los elementos pares siguientes en una lista temporaria **tmp**. Si la lista **tmp** es más larga que **subl** reemplazarla, si no seguir adelante.