

Algoritmos y Estructuras de Datos I - Año 2014 - 2do cuatrimestre  
Examen parcial - 25 de Septiembre 2014

---

**Tener en cuenta:**

- Cada ejercicio debe entregarse en **hojas separadas**, numeradas y con el nombre y apellido al lado del número de ejercicio.
- Una vez terminadas las derivaciones de un ejercicio, escribir el programa resultado final.
- Utilice el formato de derivación usado en clase.
- Sea prolijo.

1. Derivar una definición recursiva para la función especificada como

$$f.xs = \langle \exists i : 0 \leq i \leq \#xs : sum.(xs \uparrow i) = sum.(xs \downarrow i) \rangle$$

a donde *sum* calcula la suma de los elementos de una lista (puede usar la definición vista en clase).

2. Derivar una definición recursiva para la función especificada como

$$f.p.xs = \langle \text{Max } as, bs, cs : xs = as ++ bs ++ cs \wedge all.p.bs : \#bs \rangle$$

a donde *all* está definida por:

$$\begin{aligned} all.p.[ ] &\doteq True \\ all.p.(x \triangleright xs) &\doteq p.x \wedge all.p.xs \end{aligned}$$

Ayudas:

- *f* calcula la máxima longitud de segmento cuyos elementos cumplen el predicado *p*.
- No requiere generalización.

3. Especificar los siguientes problemas:

- a) Dada una lista *xs* y un número *n*, calcular la cantidad de pares de elementos de la lista cuyo producto da *n*.  
Ejemplo: con  $xs = [1, 3, 2, 4, 2]$  y  $n = 4$  el resultado es 2, ya que hay dos pares de elementos cuyo producto da 4 ( $1 * 4$  y  $2 * 2$ ).
- b) Dada una lista *xs*, calcular si hay algún segmento cuya suma es igual a la suma del resto de los elementos (esto es, los elementos que quedan fuera del segmento).  
Ejemplo: con  $xs = [1, 3, 2, 4, 2]$  sí hay, ya que el segmento  $[2, 4]$  suma 6, al igual que los elementos que quedan fuera ( $1 + 3 + 2 = 6$ ).