

**Examen Parcial - Introducción a los Algoritmos - 18 de Abril de 2016**  
**Comisiones Turno Mañana**

nota
------

1	2	3	4	5
---	---	---	---	---

**Apellido y Nombre:**

**Cantidad de hojas entregadas:** \_\_\_ (Numerar cada hoja.)

1. [10 pto(s)] Definir la función  $algunOrden : (Num, Num, Num) \rightarrow Bool$  que dada una terna de números devuelve *True* si y sólo si están en orden ascendente o descendente. Ejemplos:

(I)  $algunOrden.(10, 3, 1) = True$

(II)  $algunOrden.(1, 2, 3) = True$

(III)  $algunOrden.(1, 3, 2) = False$

2. a) [15 pto(s)] Definir la función recursiva  $absoluto : [Num] \rightarrow [Num]$ , que dada una lista de números construye una lista con los valores absolutos de los números de la primera. Ejemplo:

$$absoluto.[-2, 7, -9, -11] = [2, 7, 9, 11]$$

b) [5 pto(s)] Evaluar la función para el ejemplo anterior, justificando cada paso.

3. a) [15 pto(s)] Definir la función recursiva  $superaNota : Num \rightarrow [Num] \rightarrow Num$  que dado un número  $n$  y una lista de números  $xs$  retorna cuántos elementos de  $xs$  son mayores o iguales a  $n$ . Ejemplos:

(I)  $superaNota.4.[2, 3, 4] = 1.$

(II)  $superaNota.7.[1, 10, 2, 1, 7] = 2.$

b) [5 pto(s)] Usar la función anterior para definir la función  $regulares : [Num] \rightarrow Num$  que dada una lista  $xs$  de notas de alumnos, devuelve la cantidad de alumnos que regularizaron pero no promocionaron, es decir, la cantidad de notas  $n$  en  $xs$  que cumplen  $4 \leq n < 6$ . Ejemplos:

(I)  $regulares.[3, 5, 10, 8] = 1.$

(II)  $regulares.[4, 7, 2, 10, 6] = 2.$

4. [20 pto(s)] Dadas las siguientes funciones  $\oplus : [A] \rightarrow [A] \rightarrow [A]$  y  $\# : [A] \rightarrow Num$  definidas como:

$$\begin{aligned} [] \oplus ys &\doteq ys & \#.[] &\doteq 0 \\ (x \triangleright xs) \oplus ys &\doteq x \triangleright (xs \oplus ys) & \#.(x \triangleright xs) &\doteq 1 + \#xs \end{aligned}$$

demuestre por inducción que  $\#.(xs \oplus ys) = \#.xs + \#.ys$

5. [30 pto(s)] Dada la función recursiva  $cuantosPares : [Num] \rightarrow Num$  que cuenta la cantidad números pares en una lista, y  $soloPares : [Num] \rightarrow [Num]$  que filtra los números pares de una lista, definidas como:

$$\begin{aligned} cuantosPares.[] &\doteq 0 \\ cuantosPares.(x \triangleright xs) &\doteq \begin{aligned} &(x \bmod 2 = 0 \rightarrow 1 + cuantosPares.xs \\ &\square x \bmod 2 \neq 0 \rightarrow cuantosPares.xs \\ & \end{aligned} \end{aligned}$$

$$\begin{aligned} soloPares.[] &\doteq [] \\ soloPares.(x \triangleright xs) &\doteq \begin{aligned} &(x \bmod 2 = 0 \rightarrow x \triangleright soloPares.xs \\ &\square x \bmod 2 \neq 0 \rightarrow soloPares.xs \\ & \end{aligned} \end{aligned}$$

demuestre por inducción que  $cuantosPares.xs = \#.(soloPares.xs)$