

**Examen Parcial Introducción a los Algoritmos - 14 de Abril de 2017**  
**Comisiones Turno Tarde**

nota
------

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

**Apellido y Nombre:**

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

1. [10 pto(s)] Definir la función *distancia* :  $(Num, Num) \rightarrow Num$  que dado n par de números devuelve la diferencia entre el máximo y el mínimo. Ejemplos:

- (I) *distancia*.(1, 11) = 10
- (II) *distancia*.(2, -6) = 8
- (III) *distancia*.(-2, -5) = 3

2. (a) [15 pto(s)] Definir la función recursiva *siguientes* :  $[Num] \rightarrow [(Num, Num)]$ , que dada una lista de números retorna la lista resultante de armar un par con cada uno de ellos y sus respectivos números siguientes. Ejemplos:

(i) *siguientes*.[11, 7] = [(11, 12), (7, 8)]

(b) [5 pto(s)] Evaluar manualmente la función utilizando el ejemplo (I). Justificar cada paso.

3. (a) [15 pto(s)] Definir la función recursiva *noAparece* :  $Char \rightarrow [Char] \rightarrow Bool$  que dado un caracter *c* y una lista de caracteres (una String) *xs* retorna *True* si *c* no aparece en *xs* y *False* si está. Ejemplos:

- (I) *noAparece*.‘z’.“famaf” = *True*
- (II) *noAparece*.‘f’.“famaf” = *False*

(b) [5 pto(s)] Usar la función anterior para definir la función *soloConA* :  $[Char] \rightarrow Bool$  que dada una lista de caracteres *xs* retorna *True* si la única vocal que aparece en *xs* es ‘a’, es decir, si no aparece ninguna de las otras vocales, y *False* si aparece alguna. Ejemplos:

- (I) *soloConA*.“Famaf” = *True*
- (II) *soloConA*.“La palabra macabra” = *True*
- (III) *soloConA*.“La mar estaba serena” = *False*

4. [25 pto(s)] Dadas las siguientes funciones

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

demuestre por inducción la siguiente propiedad

$$\#(\textit{stutter}.xs) = 2 * \#xs$$

5. [25 pto(s)] Dada las siguientes funciones recursivas *cuantos* :  $Num \rightarrow [Num] \rightarrow Num$  y *agrega0si1* :  $[Num] \rightarrow [Num]$ , definidas como:

$$\begin{aligned} \textit{cuantos}.n.[ ] &\doteq 0 \\ \textit{cuantos}.n.(x \triangleright xs) &\doteq ( x = n \rightarrow 1 + \textit{cuantos}.n.xs \\ &\quad \square(x \neq n) \rightarrow \textit{cuantos}.n.xs \\ &\quad ) \end{aligned}$$

$$\begin{aligned} \textit{agregaSiguiente}.[ ] &\doteq [ ] \\ \textit{agregaSiguiente}.(x \triangleright xs) &\doteq x \triangleright ((x + 1) \triangleright \textit{agregaSiguiente}.xs) \end{aligned}$$

demuestre por inducción que  $\textit{cuantos}.1.(\textit{agregaSiguiente}.xs) = \textit{cuantos}.0.xs + \textit{cuantos}.1.xs$

(Sugerencia: Considerar los casos en que la lista empiece con 0, con 1, o con otro número diferente.)