
Nombre y apellido:
Carrera:

L.U. o D.N.I.:
Turno:

Cant. de hojas:

Departamento de Computación – FCEyN – UBA

Taller de Álgebra I - Recuperatorio

SEGUNDO CUATRIMESTRE 2018 – 21 de noviembre de 2018

Aclaraciones

- El parcial se aprueba con tres ejercicios bien resueltos.
- Programe todas las funciones en lenguaje Haskell. El código debe ser autocontenido. Si utiliza funciones que no existen en Haskell, debe programarlas. Incluya la signatura de todas las funciones que escriba.
- No está permitido alterar los tipos de datos presentados en el enunciado, ni utilizar técnicas no vistas en clase para resolver los ejercicios.

Ejercicio 1

Definir una función `mayorDistancia :: (Integer,Integer,Integer) -> Integer` que dada una tupla (a,b,c) de números enteros devuelva la máxima distancia entre dos de las coordenadas.

Por ejemplo:

`mayorDistancia (5, -2, 10) ~> 12` porque $|5 - (-2)| = 7$, $|5 - 10| = 5$ y $|-2 - 10| = 12$.

`mayorDistancia (5, 2, 10) ~> 8` porque $|5 - 2| = 3$, $|5 - 10| = 5$ y $|2 - 10| = 8$.

Ejercicio 2

Implementar una función `suma :: Integer -> Integer` que para $n \in \mathbb{N}_{>0}$ calcule la sumatoria $\sum_{i=1}^{2n} (i^2 + 2i)$.

Ejercicio 3

Programar una función `cantCambiosParidad :: [Integer] -> Integer` que dada una lista de enteros retorne cuántas veces cambia de par a impar.

Por ejemplo:

`cantCambiosParidad [1,2,3,7,6,8,7] ~> 4` (de 1 a 2, de 2 a 3, 7 a 6 y de 8 a 7).

Ejercicio 4

Programar la función `cantComunes :: [Integer] -> [Integer] -> Integer` que toma como parámetro dos listas de enteros y que retorna la cantidad de elementos de la primera lista que se encuentran también en la segunda. Si utiliza la función `elem` debe programarla.

Por ejemplo:

`cantComunes [9,1,-4,1,2] [1,-4,7] ~> 3`.

`cantComunes [9,1,-4,1,2] [-1,0,8] ~> 0`.

Ejercicio 5

Programar la función `primerasApariciones :: [Integer] -> [Integer] -> [(Integer,Integer)]` que dadas dos listas de enteros retorne una lista que contenga para cada elemento de la primera lista una tupla donde la primera coordenada es ese entero y la segunda la primera posición donde aparece en la segunda lista (0 en caso de no estar). Si utiliza la función `elem` debe programarla.

Por ejemplo:

`primerasApariciones [-4,5,1,-5,1] [9,1,-4,2,1,-4,1] ~> [(-4,3),(5,0),(1,2),(-5,0),(1,2)]`.