

Introducción a la Programación

Algoritmos y Estructuras de Datos I

Segundo cuatrimestre de 2025

Departamento de Computación - FCEyN - UBA

Práctica 5: Recursión sobre listas

Ejercicio 2.5

Implementar `quitar :: (Eq t) => t -> [t] -> [t]`, que dados un entero x y una lista xs , elimina la primera aparición de x en la lista xs (de haberla).

Posible solución Ej2.5

Posible solución

Se mostrará en clase

Ejercicio 3.3

Definir las siguientes funciones sobre listas de enteros

```
problema maximo (s: seq<ℤ>) : ℤ {  
  requiere: { |s| > 0 }  
  asegura: { resultado ∈ s ∧ todo elemento de s es menor o igual a  
             resultado }  
}
```

Posible solución Ej3.3

Posible solución

Se mostrará en clase

Ejercicio 3.9

Definir las siguientes funciones sobre listas de enteros

```
problema ordenar (s: seq<ℤ>) : seq<ℤ> {  
  requiere: { True }  
  asegura: { resultado contiene los elementos de s ordenados de  
             forma creciente }  
}
```

Sugerencia: Hay muchas formas distintas de ordenar secuencias. Una opción puede ser utilizar la función `máximo` y la función `quitar` que ya implementamos.

Posible solución Ej3.9

Posible solución

Se mostrará en clase

Ejercicio 6

Usando los siguientes tipos de datos:

- ▶ `type Texto = [Char]`
- ▶ `type Nombre = Texto`
- ▶ `type Telefono = Texto`
- ▶ `type Contacto = (Nombre, Telefono)`
- ▶ `type ContactosTel = [Contacto]`

Ejercicio 6

Usando los siguientes tipos de datos:

- ▶ `type Texto = [Char]`
- ▶ `type Nombre = Texto`
- ▶ `type Telefono = Texto`
- ▶ `type Contacto = (Nombre, Telefono)`
- ▶ `type ContactosTel = [Contacto]`

- a) Implementar una función que me diga si una persona aparece en mi lista de contactos del teléfono: `enLosContactos :: Nombre -> ContactosTel -> Bool`

Ejercicio 6

Usando los siguientes tipos de datos:

- ▶ `type Texto = [Char]`
- ▶ `type Nombre = Texto`
- ▶ `type Telefono = Texto`
- ▶ `type Contacto = (Nombre, Telefono)`
- ▶ `type ContactosTel = [Contacto]`

- Implementar una función que me diga si una persona aparece en mi lista de contactos del teléfono: `enLosContactos :: Nombre -> ContactosTel -> Bool`
- Implementar una función que agregue una nueva persona a mis contactos, si esa persona está ya en mis contactos entonces actualiza el teléfono. `agregarContacto :: Contacto -> ContactosTel -> ContactosTel`

Posible solución Ej6

Posible solución

Se mostrará en clase

Ejercicio 8

En este ejercicio vamos a trabajar con matrices.

Vamos a representar una matriz como una secuencia de secuencias.

Si m es nuestra secuencia de secuencias que representa una matriz, la secuencia i -ésima de m representa la i -ésima fila de la matriz, y el elemento j -ésimo dentro de la secuencia i -ésima representa el elemento en la fila i , columna j de la matriz.

Por ejemplo, a la matriz identidad de \mathbb{R}^3 la podemos definir como la lista de listas: $[[1, 0, 0], [0, 1, 0], [0, 0, 1]]$ en Haskell.

Usando esta representación, definir las siguientes funciones sobre matrices:

Ejercicio 8 (Similar al ejercicio 8.1)

`multiplicarFilas :: [[Integer]] -> [Integer]` según la siguiente especificación:

```
problema multiplicarFilas (m: seq⟨seq⟨ℤ⟩⟩) : seq⟨ℤ⟩ {  
  requiere: {  $|m| > 0$  }  
  requiere: {  $|m[0]| > 0$  }  
  requiere: { Todos los elementos de la secuencia  $m$  tienen la misma  
              longitud }  
  asegura: {  $|res| = |m|$  }  
  asegura: { Para todo  $0 \leq i < |res|$ ,  $res[i] = \prod_{j=0}^{|m[0]|} m[i][j]$  }  
}
```

Por ejemplo, `multiplicarFilas [[1,2],[3,4]]` es igual a `[2,12]`.

Posible solución Ej8 (Similar al 8.1)

Posible solución

Se mostrará en clase

Ejercicio 8.2

Implementar la función `cantidadDeApariciones :: Integer -> [[Integer]] -> Integer` según la siguiente especificación:

problema `cantidadDeApariciones` ($e: \mathbb{Z}$, $m: seq\langle seq\langle \mathbb{Z} \rangle \rangle$) : \mathbb{Z} {
 requiere: { $|m| > 0$ }
 requiere: { $|m[0]| > 0$ }
 requiere: { Todos los elementos de la secuencia m tienen la misma longitud }
 asegura: {
 $resultado = \sum_{i=0}^{|m|-1} \sum_{j=0}^{|m[i]|-1} 1$ si $m[i][j]$ es igual a e , 0 si no
 }
}

Por ejemplo, `cantidadDeApariciones 1 [[1,0],[0,1]]` es igual a 2.

Posible solución Ej8.2

Posible solución

Se mostrará en clase