

Estructuras Discretas

Práctica 2

Odín Miguel Escorza Soria Daniela Calderón Pérez

Facultad de Ciencias UNAM
Fecha de entrega: 7 Septiembre 2017

Realiza los siguientes ejercicios

1. Define la función **gtaveragethree** que recibe tres enteros y devuelve el número que es mayor al promedio de los tres.

Firma de la función:

`gtaveragethree :: Int → Int → Int → Int`

Ejemplos:

`*Main> gtaveragethree 1 3 6`

`6`

2. Define la función **palindromo** que nos diga si una cadena es un palindromo.

Firma de la función:

`palindromo :: String → Bool`

Ejemplo:

`*Main> palindromo "hola"`

`False`

`*Main> palindromo "oso"`

`True`

Recuerda que un **String** es una lista de caracteres en Haskell.

3. Define una función que obtenga el máximo y mínimo elemento de una lista. La función regresará un par (a,b) donde a es el mínimo y b el máximo.

Firma de la función:

`minMax :: Ord a ⇒ [a] → (a,a)`

Ejemplos

`*Main> minMax [1,2,3,4,5]`

```
(1,5)
*Main> minMax "hola"
('a','o')
```

Ten cuidado con el caso de la lista vacía, puedes usar el **error** visto en clase

4. Define una función **atN** que toma una lista y un número n y devuelve el elemento en la n-ésima posición de la lista.

Firma de la función:

```
atN :: [a] → Int → a
```

Ejemplo:

```
*Main> atN [1,2,4,6,7] 3
```

```
4
```

5. Da la definición de una función que selecciona los k primeros elementos de una lista, donde k es el elemento más pequeño de la lista.

Firma de la función:

```
selectMin :: [Int] → [Int]
```

Ejemplo:

```
*Main> selectMin [2,4,6,2,1,9]
```

```
[2]
```

¿Qué pasa cuando la longitud de la lista es menor al elemento más pequeño de la lista? Explica tu solución en el readme.

6. Da la definición de una función que borra los k primeros elementos de una lista, donde k es el elemento más pequeño de la lista.

Firma de la función:

```
deleteMin :: [Int] → [Int]
```

Ejemplo:

```
*Main> deleteMin [2,4,6,2,1,9]
```

```
[4,6,2,1,9]
```

¿Qué pasa cuando la longitud de la lista es menor al elemento más pequeño de la lista? Explica tu solución en el readme.

7. Da la función que elimine el n-ésimo elemento de una lista

Firma de la función:

```
delN :: Int → [a] → [a]
```

Ejemplo:

```
*Main> delN 4 [3,6,7,2,6,72]
```

```
[3, 6, 7, 6, 72]
```

8. Da la función que regrese un intervalo de una lista. Dados dos números y una lista regresa los elementos que se encuentran en el intervalo de esos dos números.

Firma de la función:

`intervalos :: Int → Int → [a] → [a]`

Ejemplo:

```
*Main> intervalos 2 4 [1,2,3,4,5,6,7]
[2, 3, 4]
```

9. Da la función que elimine un intervalo de una lista. Dados dos números y una lista elimina los elementos que se encuentran en el intervalo de esos dos números.

Firma de la función:

`dIntervalos :: Int → Int → [a] → [a]`

Ejemplo:

```
*Main> dIntervalos 2 4 [1,2,3,4,5,6,7]
[1, 5, 6, 7]
```

10. Da una función que diga si el promedio de los elementos de una lista es menor que la longitud de la misma.

Firma de la función:

`avgLen :: [Int] → Bool`

Ejemplo:

```
*Main> avgLen [1,2,3,14]
False
*Main> avgLen [1,2,2,1,1,1,1]
True
```

Observaciones

- Pueden usar las funciones de listas ya definidas en Haskell que vimos en clase
- Esta práctica la pueden entregar en parejas.
- El asunto de la práctica es **[ED2018-1 Práctica2]**
- Consulten los lineamientos de entrega antes de enviar
- Cualquier duda pueden mandarme correo

¡Suerte!