

Lógica Computacional 2017-1

Práctica 0

Noé Salomón Hernández Sánchez
Albert M. Orozco Camacho
C. Moisés Vázquez Reyes
Diego Murillo

Se deja el: 11 de agosto del 2016
Se entrega el: **25 de agosto del 2016**
Facultad de Ciencias UNAM

Esta práctica los introducirá al lenguaje HASKELL. En dicho lenguaje resuelva los siguientes ejercicios. No pueden usar bibliotecas que contengan funciones que se comporten como las funciones que pedimos, por ejemplo, *take*, *drop*, *etc.*

1. Ejercicios:

Considera el tipo `Nat` que representa números naturales:

```
data Nat = Cero | S Nat deriving Show
```

1. (0.5 pts) `suma :: Nat -> Nat -> Nat`

Dados dos números naturales, nos devuelve su suma.

Ejemplos:

- `*Main>suma Cero (Suc Cero)`
`Suc Cero`
- `*Main>suma (Suc Cero) (Suc $ Suc Cero)`
`Suc (Suc (Suc Cero))`

2. (0.5 pts) `prod :: Nat -> Nat -> Nat`

Dados dos números naturales, nos devuelve su producto.

Ejemplos:

- `*Main>prod Cero (Suc Cero)`
`Cero`
- `*Main>prod (Suc $ Suc Cero) (Suc $ Suc Cero)`
`Suc (Suc (Suc (Suc Cero)))`

3. (0.5 pts) `mayorQue :: Nat -> Nat -> Bool`

Dados dos números naturales, nos dice si el primero es mayor que el segundo.

Ejemplos:

- `*Main>mayorQue Cero (Suc Cero)`
`False`
- `*Main>mayorQue (Suc Cero) Cero`
`True`

4. (0.5 pts) `igual :: Nat -> Nat -> Bool`

Dados dos números naturales, nos dice si son iguales.

Ejemplos:

- `*Main>igual Cero (Suc Cero)`
`False`
- `*Main>igual (Suc Cero) (Suc Cero)`
`True`

5. (0.5 pts) `power :: Int -> Int -> Int`

Dados x, y enteros, la función debe calcular x^y

Ejemplos:

- `*Main>power 2 3`
`8`
- `*Main>power 10 0`
`1`

6. (0.5 pts) `power2 :: Int -> Int -> Int`

Implementa la potencia de enteros bajo éste esquema:

$$n^k = \begin{cases} (n^2)^{k/2} & \text{si } k \text{ es par} \\ n * (n^{k-1}) & \text{si } k \text{ es impar} \end{cases}$$

Utiliza la función `div :: Int -> Int -> Int` para hacer la división de enteros

7. (0.5 pts) `reversa :: [a] -> [a]`

Toma una lista y nos devuelve su reversa.

Ejemplos:

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

8. (0.5 pts) `sumal :: [Int] -> Int`

Toma una lista de números enteros y nos devuelve la suma de sus elementos.

Ejemplos:

- `*Main>sumal [2,3,-5,1]`
`1`
- `*Main>sumal [1,4,2]`
`7`

9. (0.5 pts) `toma :: Int -> [a] -> [a]`

Toma los primeros n elementos de una lista, $n \geq 0$

Ejemplos:

- `*Main>toma 2 [2,3,5,1]`
`[2,3]`
- `*Main>toma 10 [1,4,2]`
`[1,4,2]`

10. (0.5 pts) `tira :: Int -> [a] -> [a]`

Tira los primeros n elementos de una lista, $n \geq 0$

Ejemplos:

- `*Main>tira 2 [2,3,5,1]`
`[5,1]`
- `*Main>tira 10 [1,4,2]`
`[]`

11. (0.5 pts) `cuantas :: Eq a => a -> [a] -> Int`

Toma un elemento x y una lista ℓ , nos dice cuántas veces aparece x en ℓ

Ejemplos:

- `*Main>cuantas 1 [1,2,3,1,4]`
`2`
- `*Main>cuantas 7 [3,4,2,1]`
`0`

12. (0.5 pts) `frec :: Eq a => [a] -> [(a, Int)]`

Dada una lista ℓ , nos devuelve la siguiente lista:

$$\{(x, y) | x \in \ell, y = \text{el número de veces que aparece } x \text{ en } \ell\}$$

Ejemplos:

- `*Main>frec [1,2,3,1,4]`
`[(2,1), (3,1), (1,2), (4,1)]`
- `*Main>frec [11,8,3,1,4,4]`
`[(11,1), (8,1), (3,1), (1,1), (4,2)]`

13. (0.5 pts) `unaVez :: Eq a => [a] -> [a]`

Dada una lista ℓ , nos devuelve a los elementos que aparecen sólo una vez en ℓ

Ejemplos:

- `*Main>unaVez [1,2,3,1,4]`
`[2,3,4]`
- `*Main>unaVez [11,8,3,1,4,4]`
`[11,8,3,1]`

14. (1 pt) `compress1::String->String`

La entrada es una cadena que contiene palabras separadas por espacios, debes tomar la primer letra de cada palabra y juntarlas en una sola cadena.

Es decir, para la entrada:

Remember when I moved in you; the holy dark was moving too, and every breath we drew was Hallelujah

debes devolver:

RwImiythdwmmtaebwdwH

15. (1 pt) `compress2::String->String`

La entrada es una cadena que contiene palabras separadas por espacios, cada palabra tiene un número al inicio. De cada palabra debes obtener el caracter que esté en la posición que el número al inicio indique y debes devolverlos en una sola cadena. Si el número excede la longitud de la palabra debes devolver un espacio en blanco.

Es decir, para la entrada:

1Las 0cosas 22más 3importantes 2son 4siempre 543las 42mas 8difíciles 1de 3contar. 3Creo 8que 5eso 4es 5precisamente 193lo 3peor, 4que 8el 4secreto 1lo 23siga 2siendo, 44no 5por 1falta 0de 91un 4narrador, 3sino 2por 4falta 2de 3un 5555oyente 8comprensivo

debes devolver:

ac onp seto s r eo e ad aora i

16. (1.5 pts) `juego::(Int,Int)->(Int,Int)`

Anita y Bufarreti están jugando un juego que consiste de lo siguiente: Anita manda a Bufarreti un intervalo de números enteros $[a, b]$. Bufarreti debe responder con dos enteros, n y m , tales que:

- n es la cantidad de números en el intervalo $[a, b]$ que se pueden representar como suma de dos números primos distintos. Es decir, estos enteros cumplen:
 - $s_1 = p_{x_1} + p_{y_1}, s_2 = p_{x_2} + p_{y_2}, \dots, s_n = p_{x_n} + p_{y_n}$
 - $a \leq s_k \leq b$
 - $p_{x_k} \neq p_{y_k}$
- $m = (p_{x_1} \times p_{y_1} \times p_{x_2} \times p_{y_2} \times \dots \times p_{x_n} \times p_{y_n}) \% (10^9 + 7)$, donde $\%$ es la operación módulo.
- Ojo: Si hay más de una forma de representar a un entero como suma de dos primos, nos quedamos con la pareja cuyo producto sea menor. Es decir: si $s_k = p_{x_k} + p_{y_k}$ y $s_k = q_{x_k} + q_{y_k}$, elegiremos (p_{x_k}, p_{y_k}) si y sólo si $p_{x_k} \times p_{y_k} < q_{x_k} \times q_{y_k}$. Por ejemplo, $3+13 = 16 = 5+11$, como $3 \times 13 < 5 \times 11$, elegimos $(3,13)$.

Ejemplo:

Anita manda $(5, 12)$, entonces:

- $5 = 2+3$
- $7 = 2+5$
- $8 = 3+5$
- $9 = 2+7$
- $10 = 3+7$
- $12 = 5+7$
- $(2 \times 3 \times 2 \times 5 \times 3 \times 5 \times 2 \times 7 \times 3 \times 7 \times 5 \times 7) \% (10^9 + 7) = 9261000$

Por tanto, Bufarreti responde $(6,9261000)$.