Lenguajes de Programación
Practica # 1

15 de marzo de 2024

1. Haskell

1. Escriba una funcion `coprimos` que dados dos enteros determine si son coprimos, es decir que su *máximo común divisor* es 1.

2. Escriba una función `codifica` que dada una lista haga la compresión **RLE** o *Run-length encoding*. En ella las secuencias de datos con un mismo valor son almacenadas como un único valor con el número de repeticiones. Así,

   `codifica ‘‘aaaabccaadeeee’’`

   se obtiene

   [(4,’a’),(1,’b’),(2,’c’),(2,’a’),(1,’d’),(4,’e’)]

2. Haskell Avanzado

1. Defina un tipo de datos que guarde una fracción con sus operaciones básicas elementales **{** $+$**,** `-`**,** $/$**,** $\times$ **}** así como la operación `norm` que normaliza una fracción a la versión mas simple.

$$\texttt{norm}\frac{72}{288} \rightarrow \frac{1}{4}$$

2. Los egipcios usaban fracciones con numerador **uno** para sus numeros fraccionarios.

   Por ejemplo para denotar $\frac{3}{4}$ usaban $\frac{1}{2} + \frac{1}{4}$

   El algoritmo para conversión de una fracción a una fracción egipcia es muy sencillo:

   - Si la fracción es de la forma $\frac{1}{d}$ ya terminamos.
   - Si es de la forma $\frac{n}{d}$ entonces aproximamos por la fracción mas cercana usando $\frac{1}{\lfloor \frac{d}{n} \rfloor + 1}$ se la restamos a la original y encontramos la fracción egipcia de esta.

   Escriba una función que dada una fracción cualquiera la convierta a fracción egipcia.

3. Listas por comprehensión y comandos por guardia

1. Escriba la función *mergesort* (ordenamiento por mezcla) utilizando comandos con guardia y/o listas por comprehensión.