

Recuperatorio del trabajo práctico 2

Taller de Álgebra I

Segundo cuatrimestre 2020

1. Un *divisor propio* de un número natural $n \in \mathbb{N}$ es un divisor de n distinto de n . Llamemos $s(n)$ a la suma de los divisores propios de n . Decimos que n es *perfecto* si $n = s(n)$. Los primeros números perfectos son 6, 28, 496 y 8128.

- a) Implementar la función `sumaDeDivisoresPropios :: Int -> Int` que calcule los divisores propios positivos y los sume. Por ejemplo:

- `sumaDeDivisoresPropios 8 = 7`
- `sumaDeDivisoresPropios 12 = 16`
- `sumaDeDivisoresPropios 17 = 1`

- b) Implementar `esPerfecto :: Int -> Bool` que determina si un número es perfecto. Por ejemplo:

- `esPerfecto 1 = False`
- `esPerfecto 28 = True`
- `esPerfecto 12 = False`

2. La *sucesión alícuota* de un número n es una sucesión a_1, a_2, \dots tal que $a_1 = n$, y cada elemento a_{i+1} (para $i \geq 1$) es la suma de los divisores propios de a_i . La conjetura de Catalan–Dickson afirma que toda sucesión alícuota siempre llega a 0 o a un ciclo (conocido como *club*, ver el próximo párrafo). Hasta el momento y considerando sólo los números menores a 1000, aún no se sabe si para los números 276, 552, 564, 660 y 966 la lista es infinita o no.

Una lista de números *distintos* es un *club* (también definida como una lista de *números sociables*) si la suma de los divisores propios del k -ésimo entero coincide con el $(k+1)$ -ésimo entero y si la suma de divisores propios del último entero coincide con el valor del primer entero.

Implementar las siguientes funciones:

- a) `listaAlicuotaDeNDeLargo :: Int -> Int -> [Int]`, de modo tal que `listaAlicuotaDeNDeLargo k n` calcule los primeros k elementos de la lista alícuota de n , y
- b) `sonSociables :: [Int] -> Bool`, que determine si la lista especificada corresponde a un club.

Por ejemplo:

- `sonSociables [28] = True`
- `sonSociables [220,284] = True`
- `sonSociables [20,22,14,10] = False`
- `sonSociables [12496,14288,15472,14536,14264] = True`
- `listaAlicuotaDeNDeLargo 6 10 = [10, 8, 7, 1, 0, 0]`
- `listaAlicuotaDeNDeLargo 5 95 = [95, 25, 6, 6, 6]`

- `listaAlicuotaDeNDeLargo 4 562 = [562, 284, 220, 284]`
3. Una manera de representar a los clubes es tomando el mínimo elemento que los genera. De esta manera se puede generar una lista de clubes de cierta longitud.
- a) Implementar la función `minimosDeKClubesModuleoresQue :: Int -> Int -> [Int]` que dada una longitud k y una cota c , devuelva una lista de elementos menores o iguales a c tales que sean el elemento mínimo de un club de largo k . Por ejemplo:
- `minimosDeKClubesModuleoresQue 3 100 = []`
 - `minimosDeKClubesModuleoresQue 1 500 = [6,28,496]`
 - `minimosDeKClubesModuleoresQue 2 2000 = [220,1184]`
 - `minimosDeKClubesModuleoresQue 5 13000 = [12496]`
- b) Implementar `listaDeNClubesModuleoresQue :: Int -> Int -> [[Int]]`, una función tal que `listaDeNClubesModuleoresQue n k` devuelva una lista de todos los clubes con n miembros cuyos elementos son todos menores que k . Por ejemplo:
- `listaDeNClubesModuleoresQue 1 500 = [[6],[28],[496]]`
 - `listaDeNClubesModuleoresQue 2 2700 = [[220,284],[1184,1210]]`
 - `listaDeNClubesModuleoresQue 5 10000 = []`
4. El *índice de abundancia* de un número natural n es la suma de los divisores positivos de n dividida por n (notar que entre los divisores consideramos al mismo n). Por ejemplo, el índice de abundancia de $n = 12$ es $28/12$. Dos números se dicen *amigables* si tienen el mismo índice de abundancia. Implementar la función `esAmigosoEn :: Int -> [Int] -> Bool` de modo tal que `esAmigosoEn n l` determine si en la lista l hay algún número con el mismo índice de abundancia que n .

Condiciones de entrega. El trabajo práctico se debe hacer en forma individual. La entrega consiste de un archivo `.hs` con estas funciones implementadas, junto con todas las funciones auxiliares que sean necesarias. Las funciones pedidas en los ejercicios de este trabajo práctico deben usar los nombres especificados en cada ejercicio y respetar la signatura, dado que serán testeadas automáticamente.

El nombre del archivo entregado debe tener la forma “turno-nombreapellido.hs”, donde turno debe ser “TM”, “TT” o “TN” dependiendo del turno en el que estén cursando:

- TM: Miércoles 9 hs.
- TT: Viernes 14 hs.
- TN: Miércoles 17:30 hs.

Por ejemplo, un estudiante llamado Juan Pérez que esté cursando en el turno de los viernes debe entregar un archivo llamado “TT-JuanPerez.hs”. La entrega se debe realizar a través del campus virtual, subiendo el archivo `.hs` a través del mecanismo disponible dentro del espacio del taller en el campus virtual.

Para la resolución del trabajo práctico se deben utilizar exclusivamente los conceptos vistos hasta la semana 9 de la materia, inclusive. Se evaluará la corrección de las funciones implementadas, la declaratividad y claridad del código, y que las funciones auxiliares (si las hay) tengan nombres apropiados.

Fecha de entrega. Viernes 11/12 hasta las 22:00.