Universidade Federal de Goiás
INF - Instituto de Informática

Disciplina: Programação Funcional — Professor: Daniel Ventura
Curso: Ciências da Computação — Data: 26/09/2024

Atividade 2: **Conceitos Básicos** (data de entrega: 03/10/2024)

1. Escreva uma definição da função `intersperse` :: $a \to [a] \to [a]$ do módulo *Data.List* que intercala um valor entre os elementos duma lista. Exemplo: `intersperse` '-' "banana" = "b-a-n-a-n-a".

2. Escreva uma função `permutations` :: $[a] \to [[a]]$ para obter a lista com todas as permutações dos elementos duma lista (a ordem das permutações não é importante). Assim, se xs tem comprimento n, então `permutations` xs tem comprimento $n!$. Exemplo: `permutations` $[1,2,3] = [[1,2,3],[2,1,3],[2,3,1],[1,3,2],[3,1,2],[3,2,1]]$.

3. Ordenação de listas pelo método **merge sort**.

   (a) Defina recursivamente a função `merge` :: Ord $a \Rightarrow [a] \to [a] \to [a]$ para juntar duas listas ordenadas numa só mantendo a ordenação. Exemplo: `merge` $[3,5,7]$ $[1,2,4,6] = [1,2,3,4,5,6,7]$.

   (b) Usando a função `merge`, escreva uma definição recursiva da função `msort` :: Ord $a \Rightarrow [a] \to [a]$ que implementa o método *merge sort*:

   - uma lista vazia ou com um só elemento já está ordenada;
   - para ordenar uma lista com dois ou mais elementos, partimos em duas metades, recursivamente ordenamos as duas partes e juntamos os resultados usando `merge`.