

Universidade Federal do Ceará  
Campus de Quixadá  
Programação Funcional, 2019-1  
Primeira Avaliação  
Professor Ricardo Reis

16 de Abril de 2019

1. O algoritmo de Euler para determinação do máximo divisor comum de dois números inteiros não negativos,  $a$  e  $b$ , é dada pela função,

$$f(a, b) = \begin{cases} a & \text{se } b = 0 \\ b \mid f(a, b) & \end{cases} \quad (1)$$

desde que  $a > 0$ . O operador  $\mid$  representa resto de divisão (por exemplo,  $5 \mid 3 = 2$ ,  $13 \mid 5 = 3$ ). Implemente a função Haskell `mdc(a,n)` que determina o máximo divisor comum pelo método de Euler.

2. Considere o seguinte algoritmo de ordenação aplicado a uma lista  $\mathcal{L}$  de tamanho  $n$ .

- (a)  $i \leftarrow n - 1$ ;
- (b) Determinar o maior valor de  $\mathcal{L}$  no intervalo  $\{0, i\}$  e colocá-lo na posição  $i$ ;
- (c)  $i \leftarrow i - 1$ ;
- (d) Se  $i > 0$ , voltar para (b);

Adaptar o algoritmo anterior para o paradigma funcional implementando em Haskell a função `sort(L)` que efetua a ordenação da lista  $L$ .

3. Dada uma lista  $\mathcal{L}$  de inteiros quaisquer, construir uma função Haskell que determine a subsequência de  $\mathcal{L}$  que possua a maior soma entre todas as subsequências de  $\mathcal{L}$ .
4. Considere a aproximação,

$$\frac{4}{\pi} = 1^2 + \frac{1}{2^2 + \frac{1}{3^2 + \frac{1}{4^2 + \dots}}} \quad (2)$$

Construa função Haskell `getPi()` que determine  $\pi$  baseando-se em (2)