



## Recursão

1. Considere as seguintes relações de recorrência:

$$f(n, k) = \begin{cases} n, & \text{se } k = 1 \\ 1, & \text{se } k = n \\ f(n-1, k-1) + f(n-1, k), & \text{se } 1 < k < n \\ 0, & \text{caso contrário} \end{cases}$$

$$g(n, p, q) = \begin{cases} 2, & \text{se } n = 0 \\ p, & \text{se } n = 1 \\ p \cdot g(n-1, p, q) - q \cdot g(n-1, p, q), & \text{se } n > 1 \\ 0, & \text{caso contrário} \end{cases}$$

$$h(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2h(n-1) + 1, & \text{se } n > 1 \\ 0, & \text{caso contrário} \end{cases}$$

$$Comb(n, k) = \begin{cases} n, & \text{se } k = 1 \\ 1, & \text{se } k = n \\ Comb(n-1, k-1) + Comb(n-1, k), & \text{se } 1 < k < n \end{cases}$$

$$A(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

Implemente em Haskell essas cinco relações, considerando que as entradas são números inteiros.

2. Implemente as seguintes funções recursivas:

a. Determinar a soma dos múltiplos de 3 existentes entre 1 e N, com N dado pelo usuário. Ex:

```
> Soma_mult 10
```

```
18 (pois 18 = 3+6+9)
```

```
> Soma_mult 1
```

```
0 (pois não há múltiplos de 3 entre 1 e N quando N = 1)
```

b. Determinar o n-ésimo número de Fibonacci para  $n \geq 0$ .

Fibonacci(0) = 0.

Fibonacci(1) = 1.

Fibonacci(n) = Fibonacci(n-2) + Fibonacci(n-1)