



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BigNumbers

LEIC - Programação Funcional e em Lógica
3º Ano - 1º Semestre

Turma 7 Grupo 10

Pedro Magalhães Moreira Nunes - up201905396

Telmo Costa Botelho - up201806821

1. Fibonacci

1.1 fibRec

Implementação recursiva de fibonacci

Exemplo: fibRec 20 = 6765

1.2 fibLista

Implementação dinâmica de fibonacci

Exemplo: fibLista 20 = 6765

1.3 fibListaInfinita

Implementação com uso de listas infinitas

Exemplo: fibLista 20 = 6765

3.1 fibRecBN

Implementação recursiva de fibonacci com recurso a BigNumbers

Exemplo: fibRecBN 20 = [0,6,7,6,5]

3.2 fibListaBN

Implementação dinâmica de fibonacci com recurso a BigNumbers

Exemplo: fibListaBN 20 = [0,6,7,6,5]

3.3 fibListaInfinitaBN

Implementação com uso de listas infinitas

Exemplo: fibListaInfinitaBN 20 = [0,6,7,6,5]

2. Funções BigNumber

2.1 BigNumber

Para além da implementação pedida, o primeiro dígito do BigNumber dita o sinal do mesmo. 0 identifica um número positivo enquanto que 1 é para números negativos.

Para todas as funções é necessário inserir o número como [0,...] ou [1,...],

2.2 scanner

Transforma uma string num BigNumber

Exemplo,

scanner "23" = [0,2,3]

scanner "-23" = [1,2,3]

2.3 output

Transforma um BigNumber numa string

Exemplo,

output [0,2,3] = "23"

output [1,2,3] = "-23"

2.4 somaBN

Adiciona dois números. Se ambos tiverem sinais diferentes, serão subtraídos e o sinal será decidido pelo número de maior módulo.

Por exemplo,

$999 + 9999 = 10998$

$999 + (-9999) = -9000$

$-999 + 9999 = 9000$

$-999 + (-9999) = -10998$

```
ghci> somaBN [0,9,9,9] [0,9,9,9,9]
[0,1,0,9,9,8]
ghci> somaBN [0,9,9,9] [1,9,9,9,9]
[1,9,0,0,0]
ghci> somaBN [1,9,9,9] [0,9,9,9,9]
[0,9,0,0,0]
ghci> somaBN [1,9,9,9] [1,9,9,9,9]
[1,1,0,9,9,8]
```

2.5 subBN

Subtrai dois números. Se ambos tiverem sinais diferentes são somados

Por exemplo,

$999 - 9999 = -9000$

$999 - (-9999) = 10998$

$-999 - 9999 = -10998$

$-999 - (-9999) = 9000$

```
ghci> subBN [0,9,9,9] [0,9,9,9,9]
[1,9,0,0,0]
ghci> subBN [0,9,9,9] [1,9,9,9,9]
[0,1,0,9,9,8]
ghci> subBN [1,9,9,9] [0,9,9,9,9]
[1,1,0,9,9,8]
ghci> subBN [1,9,9,9] [1,9,9,9,9]
[0,9,0,0,0]
```

2.6 mulBN

Multiplica dois números, usando para tal duas funções auxiliares mulAux e getFactor.

Num primeiro momento com a função getFactor multiplicamos um escalar por um BigNumber. De seguida, o mulAux gera uma lista com todos os factores que são por fim somados através de foldr em mulBN.

Por exemplo,

$999 * 999 = 998001$

$999 * -999 = -998001$

$-999 * 999 = -998001$

$-999 * -999 = 998001$

```
ghci> mulBN [0,9,9,9] [0,9,9,9]
[0,9,9,8,0,0,1]
ghci> mulBN [0,9,9,9] [1,9,9,9]
[1,9,9,8,0,0,1]
ghci> mulBN [1,9,9,9] [0,9,9,9]
[1,9,9,8,0,0,1]
ghci> mulBN [1,9,9,9] [1,9,9,9]
[0,9,9,8,0,0,1]
```

2.7 divBN

Divide dois números, através de subtrações sucessivas(subBN) do número de menor valor ao número de maior valor.

Por exemplo,

$$12256440 / 29181 = (420,420)$$

```
ghci> divBN [0,1,2,2,5,6,4,4,0] [0,2,9,1,8,1]
([0,4,2,0],[0,0,0,0,0,0,4,2,0])
```

2.8 safeDivBN

Execução similar à função divBN, impedindo apenas a divisão por 0

```
ghci> a
[0,1,2,2,5,6,4,4,0]
ghci> b
[0,2,9,1,8,1]
ghci> c
[0,0]
ghci> safeDivBN a b
Just ([0,4,2,0],[0,0,0,0,0,0,4,2,0])
ghci> safeDivBN a c
Nothing
```

3. Alinea 4

Ambos Integer e BigNumber aceitam qualquer número desde que exista memória no sistema suficiente. Por outro lado, Int tem valores limitados sendo o mínimo garantido de $[-2^{29}, 2^{29}-1]$.

Quando executado com $n = 93$, as funções que usam Int dão return de um número negativo, indicando assim overflow.

fibListaInfinita 93 = -6246583658587674878

Por outro lado, utilizando o BigNumber obtemos

fibListaInfinitaBN 93 = [0,1,2,2,0,0,1,6,0,4,1,5,1,2,1,8,7,6,7,3,8]