Casos de teste

fibRec

Função recursiva simples para calculo de número de Fibonacci.

- fibRec 10 = 55 (0.00 secs)
- fibRec 20 = 6765 (0.04 secs)
- fibRec 30 = 832040 (4.42 secs)
- fibRec 35 = 9227465 (47.49 secs)

fibLista

Função recursiva com armazenamento parcial de resultados anteriores.

- fibLista 35 = 9227465 (0.00 secs)
- fibLista 100 = 354224848179261915075 (0.01 secs)
- fibLista 9999 = 2079360823(…)38230626 (0.09 secs)
- fibLista 99999 = 1605285768272981(…)278790626 (1.35 secs)
- fibLista 500000 = 2955561408(…)9780453125 (22.85 secs)

fibListaInfinita

Função com uso de lista infinita.

- fibLista 9999 = 2079360823(…)38230626 (0.06 secs)
- fibListaInfinita 99999 = 1605285768272981(…)278790626 (1.26 secs)
- fibListaInfinita 500000 = 2955561408(…)9780453125 (17.58 secs)
- fibListaInfinita 1000000 = 195328212870775(…)26838242546875 (59.13 secs)

fibRecBN

Função recursiva simples para calculo de número de Fibonacci.

- output (fibRecBN (scanner "10")) = "55" (0.00 secs)
- output (fibRecBN (scanner "20")) = "6765" (0.20 secs)
- output (fibRecBN (scanner "30")) = "832040" (23.52 secs)
- output (fibRecBN (scanner "35")) = "9227465" (282.26 secs)

fibListaBN

Função recursiva com armazenamento parcial de resultados anteriores.

- output (fibListaBN (scanner "35")) = "9227465" (0.00 secs)
- output (fibListaBN (scanner "100")) = "354224848179261915075" (0.02 secs)
- output (fibListaBN (scanner "1000")) = "43466557(…)49228875" (6.11 secs)
- output (fibListaBN (scanner "2000")) = "42246963333(…)516817125" (51.72 secs)

fibListaInfinitaBN

Função com uso de lista infinita.

- output (fibListaInfinitaBN (scanner "1000")) = "43466557(…)49228875" (6.08 secs)
- output (fibListaInfinitaBN (scanner "2000")) = "422469633(…)5168171" (50.71 secs)

scanner (com deriving Show)

Função para converter strings representando um número inteiro em BigNumber.

- scanner "123" = BN 3 (BN 2 (BN 1 EmptyList))
- scanner "-123" = Negative (BN 3 (BN 2 (BN 1 EmptyList)))
- scanner "" = Emptylist

output

Função para converter BigNumber em string.

- output (scanner "123") = "123"
- output (scanner "-123") = "-123"
- output (EmptyList) = ""

somaBN

Função que soma dois BigNumbers utilizando recursão.

Reduz sempre a uma das formas mais simples: soma de dois números positivos ou subtração de dois números positivos.

- output (somaBN (scanner "123") (scanner "123")) = "246"
- output (somaBN (scanner "123") (scanner "12")) = "135"
- output (somaBN (scanner "123") (scanner "987")) = "1110"
- output (somaBN (scanner "-123") (scanner "-987")) = "-1110"
- output (somaBN (scanner "-123") (scanner "22")) = "-101"
- output (somaBN (scanner "-123") (scanner "222")) = "99"
- output (somaBN (scanner "146") (scanner "-242")) = "-96"
- output (somaBN (scanner "146") (scanner "-24")) = "122"
- output (somaBN (scanner "123") (scanner "-123")) = "0"

subBN

Função que subtrai dois BigNumbers utilizando recursão.

Reduz sempre a uma das formas mais simples: soma de dois números positivos ou subtração de dois númeors positivos.

- output (subBN (scanner "255") (scanner "125")) = "130"
- output (subBN (scanner "100") (scanner "14")) = "86"
- output (subBN (scanner "5") (scanner "70")) = "-65"
- output (subBN (scanner "-100") (scanner "-124")) = "24"
- output (subBN (scanner "-124") (scanner "-76")) = "-48"
- output (subBN (scanner "100") (scanner "-14")) = "114"
- output (subBN (scanner "-100") (scanner "14")) = "-114"

mulBN

Funcão que multiplica 2 BigNumbers.

Reduz-se a multiplicação de 2 BigNumbers a uma série consecutiva de somas, sendo que cada argumento é calculado a partir da multiplicação de apenas um algorismo de um dos BigNumbers pelo outro BigNumber.

De forma a efetuar estes cálculos, utilizam-se 2 funções auxiliares: mulBNAux e mulBNHelper, que utilizam recursão nos cálculos.

- output (mulBN (scanner "1") (scanner "0")) = "0"
- output (mulBN (scanner "12") (scanner "0")) = "0"
- output (mulBN (scanner "123") (scanner "123")) = "15129"
- output (mulBN (scanner "-123") (scanner "123")) = "-15129"
- output (mulBN (scanner "-999") (scanner "-999")) = "998001"

divBN

Função que divide dois BigNumbers Positivos e retorna (quociente, resto).

Utiliza 2 funções auxiliares: divBNHelper e divBNInitial, que utilizam recursão.

- outputDiv (divBN (scanner "123456") (scanner "12")) = ("10288","0")
- outputDiv (divBN (scanner "123456") (scanner "13")) = ("9496","8")
- outputDiv (divBN (scanner "12") (scanner "16")) = ("0","12")

safeDivBN (com deriving Show)

Divisao que retorna o valor da divisão ou nada caso divida por zero.

- safeDivBN (scanner "17") (scanner "5") = Just (BN 3 EmptyList,BN 2 EmptyList)
- safeDivBN (scanner "17") (scanner "0") = Nothing

Alínea 4

As funções para calculo de Fibonacci utilizando Integral são, tal como esperado, mais eficientes que as utilizando BigNumber. Tal se pode verificar nos testes efetuados, resultados mostrados anteriormente, que permitem concluir que a diferença é acentuada.

Para a função fibRec conclui-se que números na casa dos 50 já resultam numa eficiência muito baixa, enquanto na fibLista o mesmo acontece na ordem de 1000000 e na fibListaInfinita na ordem dos 2000000 já se verifica uma descida acentuada na eficiência.

Para as funções utilizando BigNumbers, os números maiores aceites são significativamente mais baixos, especialmente na fibListaBN e fibListaInfinitaBN. Para a fibRecBN a diferença não é tão acentuada, sendo que perde eficiência na casa dos 40, enquanto que nas fibListaBN e fibListaInfinitaBN a perda de eficiência é similar a partir dos 2000.