

Muzyka algorytmiczna, wykład 1

Maciej Grześkowiak

3 marca 2021

Wprowadzenie do Haskell'a

Strona domowa: <http://www.haskell.org>

Instalacja: <http://www.euterpea.com/>

Kompilatory

- 1 GHC (The Glasgow Haskell Compiler)
- 2 Hugs

Obu kompilatorom towarzyszy interaktywny interpreter (ghci, hugs)

Niech

$$\text{LD}(n) = \min_a \{a : a > 1, a \mid n\}$$

Fakty

- ❶ Jeśli $n > 1$, to $\text{LD}(n)$ jest liczbą pierwszą,
- ❷ Jeśli $n > 1$ i n nie jest liczbą pierwszą, to $(\text{LD}(n))^2 \leq n$

Wniosek

n jest liczbą pierwszą jeśli $n > 1$ i $\text{LD}(n) = n$

prime0 n		n < 1 = error "liczba nie jest dodatnia"
		n == 1 = False
		otherwise = \d n == n

Przykład

Niech

$$\text{LDF}(k, n) = \min_a \{a : a \geq k, \text{LD}(n) = a\}$$

Wniosek

$$\text{LD}(n) = \text{LDF}(2, n)$$

`ldf :: Integer -> Integer -> Integer`

`ldf k n | divides k n = k`

`| k^2 > n = n`

`| otherwise = ldf (k+1) n`

`divides k n = rem n k == 0`

Zatem

```
ld :: Integer -> Integer
```

```
ld n = ldf 2 n
```

prime0 n		n < 1 = error "liczba nie jest dodatnia"
		n == 1 = False
		otherwise = ld n == n

Deklaracja typu

```
divides k n = rem n k == 0
```

```
divides : : Integer -> Integer -> Bool  
divides k n = rem n k == 0
```

1 Identyfikatory zmiennych

1 divides , rem , Id , ldf

2 Identyfikatory typów

1 True , False

Najmniejszy element listy

```
minInt :: [Int] -> Int
minInt [] = error "Pusta lista"
minInt [x] = x
minInt (x:xs) = min x (minInt xs)
```

Średnia elementów listy

```
srednia :: [Int] -> Float
srednia [] = error "Lista jest pusta"
srednia xs =
  fromIntegral (sum xs) / fromIntegral (length xs)
```

Usuwanie pierwszego znalezionej elementu a z listy

```
usunP :: Int -> [Int] -> [Int]
usunP a [] = []
usunP a (x: xs) | a == x = xs
                  | otherwise = x : usunP a xs
```
