

# Haskell 3rd tasks

Владимирова Элина

Декабрь 2022

*Задание 1.* Бесконечная упорядоченная последовательность целых чисел без повторений составлена из всех квадратов, кубов и факториалов натуральных чисел. Составить программу для вычисления  $n$ -го члена этой последовательности.

```
mergeLists :: Ord a => [a] -> [a] -> [a]
mergeLists [] ls = ls
mergeLists ls [] = ls
mergeLists ls1@(x:tail1) ls2@(y:tail2) = if x <= y
                                         then x: mergeLists tail1 ls2
                                         else y: mergeLists ls1 tail2

sqres = [x^2 | x <- [1..]]

isSquare :: Integer -> Bool
isSquare n = (k * k == n) where k = floor (sqrt (fromIntegral n))
cubes = [x^3 | x <- [1..], not (isSquare (x^3))]

fact :: [Integer]
fact' :: Integer -> [Integer] -> [Integer]
fact' n ls = (n*(last ls)) : (fact' (n+1) (ls ++ [n*(last ls)]))
fact = fact' 1 [1]

sqflist = tail (mergeLists (mergeLists sqres cubes) fact)

nSqresCubesFacts :: Integer -> [Integer]
nSqresCubesFacts n = take (fromIntegral n) (sqflist)
```

*Задание 2.* Найти первые несколько простых чисел вида  $2^n+1$ .

```
isprime :: Integer -> Bool
isprime' :: Integer -> Integer -> Bool
isprime n | n == 1      = False
          | otherwise   = isprime' n 2
isprime' n m | m*m > n   = True
          | mod n m == 0 = False
          | otherwise    = isprime' n (m+1)

cands = map (+1) (iterate (2*) 2)
```

```
simple2n1 :: [Integer]
simple2n1 = [x | x <- cands, (isprime x)]
```

*Задание 3.* Составить (бесконечный) список частичных сумм ряда, представляющего собой разложение числа  $e$ , полученное подстановкой единицы в ряд Тейлора для экспоненты.

```
fact1 :: [Double]
fact1' :: Double -> Double -> [Double] -> [Double]
fact1' n n1 ls = (1/(n*n1)) : (fact1' (n+1.0) (n*n1) (ls ++ [1/(n*n1)]))
fact1 = fact1' 1.0 1.0 [1.0]

partSums :: [Double]
partSums' :: [Double] -> Double -> [Double]
partSums' [] _ = []
partSums' (x:summands) cursum = (x + cursum) : (partSums' summands (x + cursum))
partSums = partSums' fact1 1
```

*Задание 4.* Бесконечная пирамида (см. задание 4 данной темы) содержит в узлах все последовательные нечетные числа. Составить программу для построения такой пирамиды. Какие числа будут содержаться на первых четырех уровнях этой пирамиды после добавления в нее первых пяти четных чисел? Добавление можно производить в пирамиду с помощью алгоритма «протаскивания».

```
data BinTree a = Node a (BinTree a) (BinTree a)

buildTree :: Integer -> BinTree Integer
buildTree n = Node n (buildTree (2*n + 1)) (buildTree (2*n + 3))

removeMin :: Ord a => BinTree a -> BinTree a
removeMin (Node n t1@(Node n1 _ _) t2@(Node n2 _ _)) = if n1 < n2
    then Node n1 (removeMin t1) t2
    else Node n2 t1 (removeMin t2)

insertNode :: Ord a => a -> BinTree a -> BinTree a
insertNode newn (Node n t1@(Node n1 _ _) t2@(Node n2 _ _)) | newn == n
    = Node n t1 t2
    | newn > n
    = Node n (insertNode newn t1) t2
    | newn < n
    = Node newn (insertNode n t1) t2

repeatFuncNTimes :: Int -> (a -> a) -> a -> a
repeatFuncNTimes 0 _ seed = seed
repeatFuncNTimes n f seed = repeatFuncNTimes (n-1) f (f seed)

level :: Integer -> BinTree a -> [a]
level' :: [a] -> Integer -> BinTree a -> [a]
level' s 1 (Node n _ _) = n:s
level' s i (Node _ t1 t2) = level' (level' s (i-1) t2) (i-1) t1
level = level' []
```

Тестирование:

```
main = do
  --1
  print(nSquaresCubesFacts 20)
  --2
  print(take 5 simple2n1)
  --3
  print(take 10 partSums)
  --4
  print(level 1 (buildTree 1))
  print(level 2 (buildTree 1))
  print(level 3 (buildTree 1))
  print(level 4 (buildTree 1))
  print ""
  print(level 1 (insertNode 2 (buildTree 1)))
  print(level 2 (insertNode 2 (buildTree 1)))
  print(level 3 (insertNode 2 (buildTree 1)))
  print(level 4 (insertNode 4 (buildTree 1)))
  print(level 5 (insertNode 4 (buildTree 1)))
  print ""
  print(level 1 (insertNode 4 (insertNode 2 (buildTree 1))))
  print(level 2 (insertNode 4 (insertNode 2 (buildTree 1))))
  print(level 3 (insertNode 4 (insertNode 2 (buildTree 1))))
  print(level 4 (insertNode 4 (insertNode 2 (buildTree 1))))
  print(level 5 (insertNode 5 (insertNode 2 (buildTree 1))))
  print ""
  print(level 1 (insertNode 6 (insertNode 4 (insertNode 2 (buildTree 1)))))
  print(level 2 (insertNode 6 (insertNode 4 (insertNode 2 (buildTree 1)))))
  print(level 3 (insertNode 6 (insertNode 4 (insertNode 2 (buildTree 1)))))
  print(level 4 (insertNode 6 (insertNode 4 (insertNode 2 (buildTree 1)))))
  print(level 5 (insertNode 6 (insertNode 4 (insertNode 2 (buildTree 1)))))
  print ""
```