

Практическое занятие «Хаскелл-4»: бесконечные списки

12 ноября 2019 года

Задачи данной практики должны решаться без использования рекурсии и итерации — используйте функции высших порядков. При решении следует обратить внимание на функции `concat`, `concatMap`, `group`, `groupBy`, `takeWhile`, `dropWhile` и др., а также на использование бесконечных списков.

1. Напишите функцию `gtMill :: Integer -> Integer`, принимающую натуральное число k и выдающую наименьшее натуральное число $n \geq k$, такое, что $|n/\sin(n^2)| > 10^6$.
2. Напишите функцию `recSum :: Integer -> [Integer]`, которая по заданному n строит список из n чисел t_k для $k = \overline{1, n}$, определяемых рекуррентным соотношением: $t_1 = 1$, $t_k = t_{k-1} + k \operatorname{div} 3$.
3. Напишите функцию `newton :: Int -> [Integer]`, которая принимает на вход неотрицательное целое число n и без явного вычисления факториалов выдаёт список коэффициентов бинома Ньютона степени n . Напомним, что

$$(a+b)^n = \sum_{k=0}^n C_n^k a^k b^{n-k}, \quad C_n^k = \frac{n!}{k!(n-k)!}, \quad C_{n+1}^k = C_n^k + C_n^{k+1}.$$

В последнем соотношении считаем, что $C_n^m = 0$ при $m > n$.

Ещё раз: не надо считать факториалы!!!

4. Перепишите функцию

```
root :: (Double->Double) -> Double -> Double -> Double -> Double
```

(из первой практики) поиска корня непрерывной функции на заданном отрезке методом дихотомии (двоичного деления) без использования «рукопашной» рекурсии.

5. *Знакопеременным рядом* называется последовательность $\{(-1)^n \cdot a_n\}_{n=0}^\infty$, где $a_n > 0$. Известно, что если a_n монотонно убывает к нулю при $n \rightarrow \infty$, то существует сумма бесконечного знакопеременного ряда

$$S = \sum_{n=0}^{\infty} (-1)^n \cdot a_n,$$

а k -я частичная сумма

$$S_k = \sum_{n=0}^k (-1)^n \cdot a_n$$

отличается от S не более, чем на a_{k+1} .

Напишите функцию

```
sumSerie :: (Int->Double) -> Double -> Double
```

суммирования ряда с заданной точностью. Параметрами этой функции являются функция `a :: Int -> Double`, вычисляющая значение модуля элемента ряда по его индексу, и `eps :: Double` — точность вычисления суммы. Считаем, что значение функции `a` есть положительная величина, монотонно стремящаяся к нулю с ростом значения аргумента функции.

6. Напишите функцию `allDigits :: Integer -> [Int]`, которая по заданному натуральному числу n выдаёт список цифр всех натуральных чисел от 1 до n . Например, $11 \rightarrow [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 0, 1, 1]$. Вспомогательная функция перевода числа в список цифр также должна быть написана без «рукопашной» рекурсии, а также не должна использовать функции `show` и `read`.