

Изпит по Функционално програмиране

спец. Информатика и Софтуерно инженерство, 9.02.2017 г.

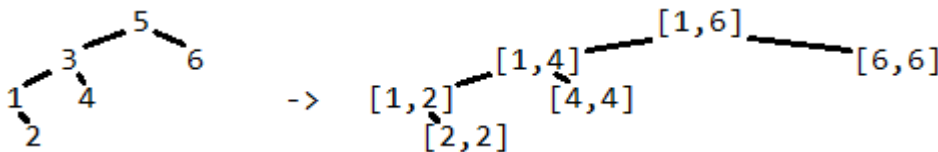
Вариант А

Задача 1 (12 т.). Да се напише функцията `largestInterval`, която получава като аргументи две едноместни целочислени функции **f** и **g** и две цели числа **a** и **b**. Функцията трябва да намира най-големия целочислен подинтервал на **[a, b]** такъв, че двете функции дават еднакви стойности за всяко цяло число в него.

Пример: `largestInterval (\x -> x) (\x -> x*x) 0 3 -> (0, 1)`

Задача 2. (16 т.) Да се напише функцията `intervalTree`, която преобразува двоично дърво от числа в ново дърво със същата структура, в което стойността във всеки възел е заменена с наредена двойка, представляваща най-малкия интервал, съдържащ всички стойности в съответното поддърво. Представянето на дървото е по

Пример:



Бонус (4 т.): `intervalTree` да работи за време $O(n)$ в най-лошия случай.

Задача 3. (10 т.) Да се генерира поток `sumsOfSquares` от тези числа, които са сума от квадратите на две положителни цели числа.

Задача 4. (12 т.) Видеоклип се представя с име (низ) и дължина (брой секунди). Да се напише функцията `averageVideo`, която по непразен списък от видеоклипове намира името на този, който е с дължина най-близка до средната дължина на всички видеоклипове в списъка, без да я надхвърля.

Пример: `averageVideo [("lolcat", 15), ("dogewow", 35), ("omgseethis", 28)] -> "lolcat"`

Изпит по Функционално програмиране

спец. Информатика и Софтуерно инженерство, 9.02.2017 г.

Вариант Б

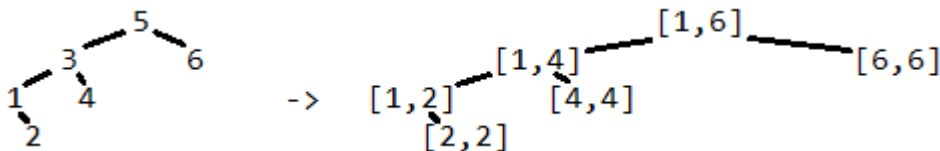
Задача 1. (12 т.) Да се напише функция `countIntervals`, която получава като аргументи две едноместни целочислени функции **f** и **g** и две цели числа **a** и **b**. Функцията трябва да намира броя на всички непразни целочислени подинтервали на **[a, b]**, в които да няма цяло число, за което функциите да дават еднакви стойности.

Пример: `countIntervals (\x -> x) (\x -> x*x) 0 3 -> 3`

Интервалите са `[2, 2]`, `[2, 3]` и `[3, 3]`.

Задача 2. (16 т.) Да се напише функция `pairTree`, която преобразува двоично дърво от числа в ново дърво със същата структура, в което стойността във всеки възел е заменена с наредена двойка, представляваща най-малката стойност в лявото поддърво на съответния възел (включително и самия възел) и най-голямата стойност в дясното поддърво на съответния възел (включително и самия възел).

Пример:



Бонус (4 т.): `pairTree` да работи за време $O(n)$ в най-лошия случай.

Задача 3. (10 т.) Да се генерира поток `sumsOfCubes` от тези числа, които са сума от кубовете на две положителни цели числа.

Задача 4. (12 т.) Чифт обувки се представят с модел (низ) и номер (цяло число). Да се напише функция `bestRange`, която по даден непразен списък от чифтове намира модел обувки, от който има максимален брой различни номера.

Пример: `bestRange [("boots", 38), ("sandals", 41), ("boots", 38), ("sandals", 43)] -> "sandals"`