

Project Euler Task 188

Козиний Сергей

7 октября 2016 г.

Problem 188

“The hyperexponentiation or tetration of a number a by a positive integer b , denoted by $a \uparrow\uparrow b$ or ${}^b a$, is recursively defined by:

$$a \uparrow\uparrow 1 = a, a \uparrow\uparrow (k+1) = a^{(a \uparrow\uparrow k)}.$$

Thus we have e.g. $3 \uparrow\uparrow 2 = 3^3 = 27$, hence $3 \uparrow\uparrow 3 = 3^{27} = 7625597484987$ and $3 \uparrow\uparrow 4$ is roughly $(10^{3.6383346400240996 \cdot 10^{12}})$.

Find the last 8 digits of $1777 \uparrow\uparrow 1855$.”

Главная и единственная проблема возникающая при решении Problem 188 - проблема памяти. Размещение в памяти чисел $3 \uparrow\uparrow 5$ и больших представляется невозможным. Но поскольку нам необходимо получить только 8 последних цифр числа $1777 \uparrow\uparrow 1855$, то для решения проблемы, можно обратиться к свойствам операции *mod*:

$$\begin{aligned}(a+b) \bmod c &= ((a \bmod c) + (b \bmod c)) \bmod c \\ (a*b) \bmod c &= (a \bmod c) * (b \bmod c) \bmod c \\ a^b \bmod c &= (a \bmod c)^b \bmod c\end{aligned}$$

Определим операцию *hyperExpMod* как рекурсивную функцию трёх аргументов (так как она была дана в тексте задания)

```
— Haskell lang
hyperExpMod base 0 m = 1
hyperExpMod base pow m =
    expMod base (hyperExpMod base (pow-1) m) m
```

Вспомогательная операция *expMod* решает проблему получения остатка от деления a^b на m и использует модифицированный алгоритм бинарного (быстрого) возведения в степень. Модификация заключается в том, что на каждой итерации *expMod* от основания b и аккумулятора p берётся остаток от деления на m (что имеем право делать по свойствам операции *mod*). Этим гарантируется то, что при рекурсивном вызове *expMod*, на каждой итерации, b и p будут меньше чем m , а следовательно уместятся в память.

```
— Haskell lang
expMod b p m = expModIter 1 b p m
    expModIter res b p m
        | p==0 = res `mod` m
        | p `mod` 2 == 0 = expModIter res ((b^2) `mod` m) (div p 2) m
        | otherwise = expModIter (res*b `mod` m) (b `mod` m) (p-1) m
```

Поскольку нам необходимо найти последние восемь цифр данного числа, целесообразно воспользоваться типом *Integer* который обеспечивает работу с длинной арифметикой (на каждой из итераций *expMod*, *mod m* будет вычисляться от произведения чисел каждое из которых $\leq 10^8$, а значит левый аргумент *mod* будет $\leq 10^{16}$ что может не уместиться в обычный *Int*). Для этого достаточно указать сигнатуру типа для результирующего значения

```

-- Haskell lang
task188 :: Integer
task188 = hyperExpMod 1777 1855 100000000
  where
    hyperExpMod base 0 m = 1
    hyperExpMod base pow m =
      expMod base (hyperExpMod base (pow-1) m) m
    expMod b p m = expModIter 1 b p m
    expModIter res b p m
      | p==0 = res `mod` m
      | p `mod` 2 == 0 = expModIter res ((b^2)`mod`m) (div p 2) m
      | otherwise = expModIter (res*b `mod` m) (b`mod`m) (p-1) m

```

Результат:

```

#Console
austrotaxus@small-box:~/EulerProj$ stack ghci
Using main module: Package 'EulerProj' component exe:EulerProj-exe
with main-is file: /home/austrotaxus/EulerProj/app/Main.hs
The following GHC options are incompatible with GHCi and have not
been passed to it: -threaded
Configuring GHCi with the following packages: EulerProj
GHCi, version 7.10.3: http://www.haskell.org/ghc/ :? for help
[1 of 2] Compiling Lib
( /home/austrotaxus/EulerProj/src/Lib.hs, interpreted )
[2 of 2] Compiling Main
( /home/austrotaxus/EulerProj/app/Main.hs, interpreted )
Ok, modules loaded: Lib, Main.
*Main Lib> task188
95962097

```