Домашна работа по "Функционално програмиране"

Васил Даниелов Пашов, СИ, IV курс, ФН: 61781

1 Задача 2 (бонус)

Нека е дадено неотрицателно цяло число n. Напишете функция (reduce n), която го "редуцира"до едноцифрено по следната процедура:

1. намира най-голямата цифра в числото и я "премахва"от него (при повече от едно срещания премахва най-лявата такава цифра)
2. умножава новополученото число по тази премахната цифра и, ако полученото число не е едноцифрено, повтаря процедурата наново за него.

Искаме да докажем, че процедурата приключва за всяко неотрицателно цяло число. За тази цел първо ще докажем следното твърдение:

Твърдение 1.0.1. *Нека n е неотрицателно цяло число, числото получено след премахване на най - лявото срещане на най - голямата цифра на n ще бележим с n', тогава за всяко n, n' е поне 10 пъти по - малко от n.*

Доказателство. Нека разгледаме представянето на n и n' в десетична бройна система, и нека a_k да бъде най - голямата цифра в n:

$$n = a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_{k+1} 10^{k+1} + a_k 10^k + a_{k-1} 10^{k-1} \ldots a_1 10 + a_0$$
$$n' = a_n 10^{n-1} + a_{n-1} 10^{n-2} + \cdots + a_{k+1} 10^k + a_{k-1} 10^{k-1} \ldots a_1 10 + a_0$$

Искаме да покажем, че $10n' < n$ т.е. $n - 10n' > 0$. Нека разгледаме какво се получава, когато извадим двете:

$$\begin{aligned} n - 10n' = {} & a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_{k+1} 10^{k+1} + a_k 10^k + a_{k-1} 10^{k-1} \ldots a_1 10 + a_0 \\ & -(a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_{k+1} 10^{k+1} + a_{k-1} 10^k \ldots a_1 10^2 + a_0 10 + 0) \end{aligned}$$

Очевидно членовете с индекси $n, n-1 \ldots k+1$ се унищожават и оставаме с:

$$n - 10n' = a_k 10^k + a_{k-1} 10^{k-1} \ldots a_1 10 + a_0 - (a_{k-1} 10^k \ldots a_1 10^2 + a_0 10 + 0)$$

Сега за да преценим дали резултатът е положителен е достатъчно да сравним почленно коефицитентите пред съответните степени на 10. Нека отбележим, че $\forall a_i \in \{a_0, a_1, \ldots a_{k-1}\} : a_k \geq a_i$ и освен това $a_k \neq 0$. Ако $a_k > a_{k-1}$ е очевидно, че $n > n'$, иначе е задължително $a_{k-1} = a_k$, по същия начин продължаваме със следващите членове a_{k-1} (от n) и a_{k-2} (от n'). И така има два варианта в някой момент ще достигнем до a_i (от n'), за което $a_{i+1} = a_k > a_i$, и ще обавим $n > n'$ или ще достигнем до момента, в който сравняваме a_0 и 0, но тогава $a_0 = a_k > 0$.

□

Сега вече можем да докажем следното твърдение:

Твърдение 1.0.2. *Описаната процедура завършва за краен брой стъпки*

Доказателство. Нека разгледаме редицата $n, n_1, n_2 \ldots n_p$. За всеки член n_i на редицата знаем, че $n_i = n'_{i-1} * a_k, a_k \in [1; 9]$, следователно от Твърдение 1.0.1 $n_i > n_{i+1} > \cdots > n_k$ и от това, че разглеждаме *цели* неотрицателни числа следва, че редицата намалява с поне единица на всяка стъпка. Тогава е сигурно, че в някой момент $10 > a_p$, и алгоритъмът ще приключи. □