CSL765: Introduction to (Logic and Functional) Programming
I semester 2019-20
Assignment 1.

Karatsuba's Algorithm and Factorials

1 Multiplication of large positive integers

The multiplication of two n-digit numbers by the usual long multiplication algorithm (column-wise) that is taught in schools takes about $O(n^2)$ single-digit multiplications.

According to the Wikipedia, Karatsuba's algorithm (discovered by Anatoly Karatsuba in 1960) reduces the complexity of multiplication of two n-digit numbers to at most $n^{lg_2 3}$ single-digit multiplications.

2 Karatsuba's algorithm

Given two n-digit (for large n) positive integers x and y in a base $B > 1$. For $m = \lceil n/2 \rceil = \lfloor (n+1)/2 \rfloor$ we have

$$\begin{array}{lcll} x & = & x_1 B^m + x_0, & \text{for some } x_0, x_1 < B^m \\ y & = & y_1 B^m + y_0, & \text{for some } y_0, y_1 < B^m \\ x.y & = & x_1 y_1 B^{2m} + (x_1 y_0 + x_0 y_1) B^m + x_0 y_0, & \text{for some } y_0, y_1 < B^m \end{array}$$

Let

$$\begin{array}{lcll} z_2 & = & x_1 y_1 & < B^{2m} \\ z_1 & = & x_1 y_0 + x_0 y_1 & < 2B^{2m} \\ z_0 & = & x_0 y_0 & < B^{2m} \end{array}$$

Hence

$$xy = z_2 B^{2m} + z_1 B^m + z_0 \tag{1}$$

would require 4 multiplications of m-digit numbers of which z_1 requires two multiplication operations. Hence for any $k > 0$, such that $2^{k-1} < n \leq 2^k$, the computation of xy would require $O(n^2)$ multiplications.

Now consider

$$\begin{array}{lcl} & & (x_1 + x_0).(y_1 + y_0) \\ & = & z_2 + z_1 + z_0 \end{array}$$

It is possible that the computation of z_1 may result in an overflow.

$$\begin{array}{lcl} z_1 & = & (x_1 + x_0).(y_1 + y_0) - z_2 - z_0 \\ & = & (x_1 + x_0).(y_1 + y_0) - x_1 y_1 - x_0 y_0 \\ & = & (x_1 y_1 + x_1 y_0 + x_0 y_1 + x_0 y_0 - x_1 y_1 - x_0 y_0 \\ & = & (-x_1 y_1 + x_1 y_0 + x_0 y_1 - x_0 y_0) + x_1 y_1 + x_0 y_0 \\ & = & -(x_1 y_1 - x_1 y_0 - x_0 y_1 + x_0 y_0) + x_1 y_1 + x_0 y_0 \\ & = & -(x_1 - x_0)(y_1 - y_0) + x_1 y_1 + x_0 y_0 \\ & = & (x_0 - x_1)(y_1 - y_0) + x_1 y_1 + x_0 y_0 \\ & = & sgn(x_0 - x_1).sgn(y_1 - y_0).|x_0 - x_1|.|y_1 - y_0| + x_1 y_1 + x_0 y_0 \end{array}$$

where $sgn(x)$ denotes the sign of x. Now the computation of z_1 requires only one multiplication operation though we may have to deal with negative numbers because of the subtraction operations. The numbers $(x_0 - x_1)$ and $(y_1 - y_0)$ lie in the interval $[-B^m, B^m]$. However since $|x_0 - x_1| \leq max(x_1, x_0)$ and $|y_1 - y_0| \leq max(y_1, y_0)$ By separating out the signs and doing an absolute value multiplication, we are guaranteed that $|x_0 - x_1|.|y_1 - y_0| \leq max(x_1, x_0) \times max(y_0, y_1)$ will not overflow if none of the other products computed in z_2 and z_0 do not overflow.

Problem Statement

Your task is to implement a function to compute factorial of large numbers using Karatsuba's multiplication algorithm in Standard ML or OCaml.

What you have to do

1. Decide whether to program in Standard ML or OCaml and choose the appropriate instruction from each of the following steps.
2. Create a file `<entry-number>.ml` if you choose to use OCaml or `<entry-number>.sml` if you choose Standard ML.
3. Note that the input numbers to your program might be larger than `Int.maxInt` (in case of sml) or `max_int` (in case of OCaml).
4. You are NOT allowed to use existing libraries and structures (such as `IntInf` or `Big_int`) for arbitrarily large integers.
5. Your submission should contain two methods - `karatsuba` (implements Karatsuba's multiplication) and `factorial` (computes factorial using Karatsuba's method wherever multiplication is required). We might evaluate both of these methods separately.
6. The input and output numbers to these functions are strings of digits.
7. Define functions `fromString : string -> int list` and `toString: int list -> string` for purposes of input and output respectively.
8. You have to manipulate the numbers as lists of digits in base $B = 10^9$ (for OCaml) and $B = 10^4$ (for Standard ML).
9. The Karatsuba multiplication function takes two numbers as input and returns the result of multiplication. The function to compute Karatsuba multiplication should have signature `val karatsuba: int list -> int list -> int list`.
10. The factorial function takes one *non-negative* integer and returns it's factorial. The function should have signature `val factorial: string -> string`.
11. If the input string is not a valid number, or the input to factorial is negative your program should raise an exception of type `exception Invalid_Input_exception of string`

Submission Instruction

Submit an Ocaml file `<entry-number>.ml` or a Standard ML file `<entry-number>.sml` on Moodle (`https://moodle.iitd.ac.`

Important Notes

1. Do not change any of the names given in the signature. You are not even allowed to change upper-case letters to lower-case letters or vice-versa.
2. You may define any new functions you like besides those mentioned in the signature.
3. **Do not use libraries for arbitrary precision arithmetic.**

4. Follow the input output specification as given. We will be using automated script for evaluation. In case of mismatch, you might be awarded zero marks.

5. Since the evaluator is going to look at your source code before evaluating it, you must explain your algorithms in the form of ML comments, so that the evaluator can understand what you have implemented.