2WF90: Software Assignment Integer Arithmetic

Andreas Hülsing, Benne de Weger

September 2021

Contents

1 Introduction

The programming assignment for the first part of the course consists of writing code ('from scratch') for arithmetic with large integers, including modular arithmetic. This is also meant as preparation for the second programming assignment on polynomial and finite field arithmetic, which is described in a separate document.

1.1 Role in assessment

This programming assignment contributes $7\frac{1}{2}\%$ to your 2WF90 grade.

1.2 General guidelines for the programming assignments

There is a separate document with "General Guidelines", that also apply for this assignment.

2 Software Assignment on Integer Arithmetic

Write code 'from scratch' for basic arithmetic with large integers using representations with as radix a configurable integer $b \leq 16$). 'From scratch' means that you are not allowed to use large-integer libraries built into Python, but your code should use from Python only elementary arithmetic operations on small integers of at most 32 bits.

Your code should be able to do the following:

- formats, input, output:
  - deal with the standard radix b representation of integers of large word length (at least several hundreds); word length should be variable;

  - compute directly with the radix b, i.e. conversion back and forth to another radix is not allowed;
  - inputs and outputs of arithmetic operations should be in this standard format; your code should support any input and output files with the required format, and your code may assume that the provided input files indeed do follow this format, so do not spend any time on writing error handling code for possibly erroneous input;
  - deal with positive and negative integers, and with zero;

- Integer arithmetic:
  - the following integer arithmetic operations should be supported:
    * addition, subtraction;
    * multiplication by the primary school method;
    * multiplication by the method of Karatsuba (dividing up long numbers in two parts, with recursion);
    * Euclid's Extended Algorithm for two large positive integers;
  - for the multiplication operations, the software should keep track of the number of elementary additions/subtractions it does, and the number of elementary multiplications, and to be able to show those operation counts in the output.

- Modular arithmetic:
  - modular arithmetic with a modulus m of large word length: the following modular arithmetic operations should be supported:
    * modular reduction (i.e. "long" division by m with remainder, where only the remainder is relevant output; see Algorithm 1.5);
    * addition, subtraction (mod m);
    * multiplication (mod m) (primary school multiplication method suffices);
    * modular inversion for a large modulus m.

Present some examples of different word sizes and radices, and compare elementary operation counts of primary school and Karatsuba methods. Draw a conclusion based on the theory. Input and output files are not to be submitted.

In grading your program will be tested with a set of test exercises that will stay unknown to you.