CISC 822 Algebraic Algorithms Fall 1998
First Exercise Set Due Wed, Oct 21

Collins & Saunders

October 20, 1998

1. Prove the first 4 statements of Theorem 1 of Collins, *The computing time of the Euclidean algorithm*, JACM 3, 1974. This paper was handed out in class. Let f, f', g, g' be non-negative real-valued functions, and let c be a positive real number. The four statements to prove are:

   (a) $f \sim cf$.

   (b) if $f \preceq g$ and $f' \preceq g'$, then $f + f' \preceq g + g'$ and $ff' \preceq gg'$.

   (c) $f \preceq g$ and $f' \preceq g$ implies $f + f' \preceq g$.

   (d) $\max(f, g) \sim f + g$.

2. This question relates to Knuth section 4.3.3 exercise 13. The exercise is about multiplying an m bit number A by an n bit number B, when n is substantially larger than m. An approach is given in the answers section in the back of the book along with an analysis of it's run time. The basic idea of that method is to split B into m bit parts and multiply each part by A. The method implemented in the `KARATSUBA` procedure in saclib is a bit different. It splits B into two n/2 bit numbers and recursively calls itself on the two parts (see `Step2` of `KARATSUBA`).

   (a) Show this method has running time codominant with that given in Knuth's book.

   (b) Which method do you think is better in practice? why?

3. Section 4.5.1 Exercise 6.

4. Section 4.5.2 Exercise 17.

5. Saclib has representations for a number of mathematical domains including integers, rational numbers, and integer coefficient polynomials in one or more variables. All of these domains contain a zero element, and in all of the representations it is represented by the zero Word (all bits zero). All of these domains also contain the integers in a natural way. For instance

rational numbers with denominator 1 are integers, constant polynomials are integers. Yet the representation of an integer differs in each of these saclib representations. For example, consider 2. In the rationals it is (2, 1), in the univariate integer polynomials it is (0, 2), in the bivariate polynomials it is (0, (0, 2)), etc.

Explain why those rational numbers which are integers are not simply represented as they are in the integers. (Show how ambiguities would arise, which would confuse the rational arithmetic procedures.) Sketch a design of a system supporting arbitrary integers, rationals, polynomials, in which the representation of all integers is like the representation of zero in saclib: the same, regardless of the domain in which it is embedded. What are some pros and cons of your design relative to saclib's? This is an essay question; the expected answer length is a page or so .

6. Write and test a program to meet this specification:

```
/*===========================================================================
 n <-- RLOG2(R,k; n,f)

Rational number logarithm, base 2.

Inputs
  R : a positive rational number,
  k : a positive BETA-digit.

Output
  n : a bit BETA-digit, the integer part of L, where L is log_2(R).
  f : a BETA-digit such that f/2^k <= L < (f+1)/2^k.
===========================================================================*/
#include "saclib.h"

void RLOG2(R,k, n_, f_)
       Word R,k, *n_,*f_;
{
   ///
}
```

7. The first `for` loop in IPEMV could be replaced by

```
 /* assuming "Word ap;" is declared above in the code  */
 ap = IEXP(a, e1 - e2)
 if (rp == 0)
   B = IPROD(ap,B);
 else
   B = IPIP(rp,ap,B);
```

And the second `for` loop could be similarly replaced.

The idea of the change is to compute $(a^k)B$ rather than $(a(a \ldots (aB) \ldots))$, for $k = e1 - e2$, a conjectured advantage if B is an extensive polynomial.

(a) Do an analysis to produce a good dominating function, preferably codominant function, for the cost of the algorithm with and without this change. Compare. Measure the cost in terms of the degree of the polynomial in the main variable and the total number of terms.

(b) implement the change and test the performance with and without the change on a variety of inputs to IPEMV.