# MATH 895, Assignment 2, Summer 2009

Instructor: Michael Monagan

Please hand in the assignment by 3:30pm on June 10th by the beginning of class.
Late Penalty -20% off for up to 24 hours late. Zero after that.
Please submit a printout of a Maple worksheet containing Maple code and output.
This assignment and the next are heavy on programming. The last two won't be.

Download and read the paper "Polynomial Division using Dynamic Arrays, Heaps and Packed Exponent Vectors" by Monagan and Pearce from

`http://www.cecm.sfu.ca/personal/monaganm/teaching/TopicsinCA09/`

The goal of this assignment is to study the data structures considered in the paper for polynomial division using a sparse distributed representation.

## 1 Fixing Multiplication.

Let $f, g \in R[x_1, x_2, ..., x_n]$ where $f = f_1 + f_2 + ... + f_{\#f}$ and $g = g_1 + g_2 + ... + g_{\#g}$. To compute $h = f \times g$ we suggested using the following divide an conquer approach, namely, if $\#f > 1$ compute

$$A = (f_1 + f_2 + ... + f_k) \times g \quad \text{and} \quad B = (f_{k+1} + ... + f_{\#f}) \times g \quad \text{where} \quad k = \lfloor \#f/2 \rfloor$$

recursively then add $A + B$ using a merge. Determine the number of comparisons this algorithm does in the worst case. Express your answer in big O notation in terms of $\#f$ and $\#g$.

## 2 Fixing Division.

Let $A, B \in \mathbb{Q}[x, y, z, ...]$ and let $Q$ be the quotient of $A$ divided $B$. Represent a polynomial as a Maple list of terms sorted in descending *graded lexicographical* order. Represent each term in the form $[c, e]$ where $c \in \mathbb{Q}$ is a coefficient and $e$, the exponent vector, is encoded as an integer. E.g. the monomial $x^i y^j z^k$ with exponent vector $[i, j, k]$ would be represented as the integer $(i + j + k)B^2 + iB + j$ where $B = 2^L$ bounds the total degree $d$ of any monomial that appears in the division algorithm. Implement the following Maple procedures where $X$ is a list of variables.

```
SDMP2Maple(a,X,d)
Maple2SDMP(A,X)
```

E.g. `A := SDMP2Maple(a,[x,y,z],d)` converts a Maple polynomial $a(x, y, z)$ into the SDMP data structure and `Maple2SDMP(A,[x,y,z])` converts it back. Note, to convert an integer E to base B in Maple use `convert(E,base,B);`

Now design and implement 1 and any two of (2,3,4) algorithms below, or, for a bonus, all four algorithms.

1 Johnson's heap multiplication algorithm: $f \times g = \sum_{i=1}^{\#f} f_i \times g$,

2 division using the repeated merging algorithm: $((f - q_1 \times g) - q_2 \times g) - q_3 \times g - ...$

3 Johnson's quotient heap division algorithm: $f - \sum_{i=1}^{\#q} q_i \times g$,

4 Monagan and Pearce's divisor heap division algorithm: $f - \sum_{i=1}^{\#g} g_i \times q$.

For the heap algorithms you may use Maple's `heap` package. See `?heap`. Execute your algorithm on the following sparse problem using your distributed data structure for each algorithm.

```
> X := [u,v,w,x,y,z];
> a := randpoly(X,degree=10,terms=2500):
> b := randpoly(X,degree=5,terms=8):
> c := expand(a*b):
> nops(a), nops(b), nops(c);
                                    2479, 8, 16401

> d := degree(a)+degree(b);
> A := Maple2SDMP(a,X,d):
> B := Maple2SDMP(b,X,d); # show your data structure for this one
> C := Maple2SDMP(c,X,d):
> H := MULTIPLY(A,B): evalb(H=C); # Johnson's heap multiplication
> H := MULTIPLY(B,A): evalb(H=C);
> Q := DIVIDE(C,A): evalb(Q=B);
> Q := DIVIDE(C,B): evalb(Q=A);
```

Compute and print (i) $N =$ the number of monomial comparisons each algorithm makes, (ii) $M =$ the number of coefficient multiplications + divisions each algorithm makes and (iii) the quantity $S = N/M$ which measures the monomial comparisons relative to the coefficient arithmetic cost. In a few sentences discuss whether the results agree with the theoretical cost estimates of the algorithms.