# Assignment 2
# CS 9566A

Paul Vrbik
250389673

October 14, 2009

## Question 1

```
RED:=proc(A,B,x)
local m,n,Ap;

    printf("Call RED(%a,%a,x)\n",A,B);
    m,n:=degree(A,x),degree(B,x);
    if m<n then
        return A;
    else
        Ap:=expand(A - coeff(A,x,m)*x^(m-n)*B);
        return RED(Ap,B,x);
    end if;

end proc;
```

```
> f:=3+x-x^2+x^3:
> g:=x-1:
> rem(f,g,x);
                                    4
> RED(f,g,x);
Call RED(3+x-x^2+x^3,x-1,x)
Call RED(3+x,x-1,x)
Call RED(4,x-1,x)
                                    4
```

## Question 2

```
fastRED:=proc(A,B,x)
local m,n,Bs,S,As,Qs,q,r;

    m,n:=degree(A,x),degree(B,x);

    if m<n then
        return A;
    end if;

    Bs:=add( coeff(B,x,i)*y^(n-i), i=0..n);
    As:=add( coeff(A,x,i)*y^(m-i), i=0..m);

    gcdex(Bs,y^(m-n+1),y,'S','t');
#   => S*Bs + t*y^(m-n+1) = 1 => S*Bs = 1 mod y^(m-n+1)
```

```

    Qs:=rem( expand(As*S), y^(m-n+1), y);

    q:=add( x^(m-n-i)*coeff(Qs,y,i), i=0..m-n );
    r:=expand(A-B*q);

    return (q,r);
end proc:
```

```
> f:=3+x-x^2+x^3;
> g:=x-1;
> trace(fastRED):
> fastRED(f,g,x);
{--> enter fastRED, args = 3+x-x^2+x^3, x-1, x
                               m, n := 3, 1

                               Bs := 1 - y

                                      3    2
                         As := 1 + 3 y  + y  - y

                                    1

                                      2
                               Qs := y  + 1

                                        2
                               q := 1 + x

                                  r := 4

<-- exit fastRED (now at top level) = 1+x^2, 4}
                                     2
                                1 + x , 4
```

## Question 3 - Power Series Root

Let $F = 1 + f_1x + f_2x^2 + \cdots$ and $G = 1 + g_1x + g_2x^2$, to find $G$ such that $G^2 = F$ we do

$$\begin{aligned} G^2 = g_0^2 \\ &+ (g_0g_1 + g_1g_0)x \\ &+ (g_0g_2 + g_1g_1 + g_2g_0)x^2 \\ &+ (g_0g_3 + g_1g_2 + g_2g_1 + g_3g_0)x^3 \\ &+ \cdots \\ &+ \left(\sum_{k=0}^{n} g_kg_{n-k}\right) x^n. \end{aligned}$$

Using this general pattern to do coefficient matching we find

$$\begin{aligned} g_0^2 = 1 &\Rightarrow g_0 = 1 \\ 2g_0g_1 = f_1 &\Rightarrow g_1 = f_1/2 \\ 2g_0g_2 + g_1^2 = f_2 &\Rightarrow g_2 = \frac{1}{2}(f_2 - f_1^2) \\ &\vdots \\ f_n = \sum_{k=0}^{n} g_kg_{n-k} &\Rightarrow g_0g_n = g_n = \frac{1}{2}\left(f_n - \sum_{k=1}^{n-1} g_kg_{n-k}\right) \qquad \text{(for } n > 1\text{).} \end{aligned}$$

For a simple induction argument notice that the reduction above shows that the first three terms of $g$ are uniquely determined by $f$ (up to a sign change). If we assume that $g_{k-1}$ is uniquely determined by terms of $f$ then $g_k$ is uniquely determined by $f$ as well because

$$2g_0g_n + \sum_{k=1}^{n-1} g_kg_{n-k} = f_n \tag{1}$$

$$g_n = \frac{1}{2}\left(f_n - \sum_{k=1}^{n-1} g_kg_{n-k}\right) \tag{2}$$

(left hand side of (2) is uniquely determined since each term of the difference is uniquely determined).

For the complexity we first observe that

$$\begin{aligned} \sum_{k=0}^{n} g_kg_{n-k} &= 2\sum_{k=0}^{n/2-1} g_kg_{n-k} + g_{n/2}^2 && n \text{ even} \\ \sum_{k=0}^{n} g_kg_{n-k} &= 2\sum_{k=0}^{n/2-1} g_kg_{n-k} && n \text{ odd.} \end{aligned}$$

In either case we require $O(n/2)$ $\times$'s and $+$'s for $g_n$. Therefore to get $n$ terms of $g$ requires

$$\sum_{i=0}^{n} O(i/2) = \frac{(n/2)(n/2+1)}{2} = O(n^2)$$

$\times$'s and $+$'s.

## Question 4

If we let $F = 1 + 2x$ in Question 3 we can use formula (2) to build the first ten terms of $G$ (where $G = \sqrt{F}$). A simple Maple program (omitted) gives:

$$G = 1 + x - \frac{1}{2}x^2 + \frac{1}{2}x^3 - \frac{5}{8}x^4 + \frac{7}{8}x^5 - \frac{21}{16}x^6 + \frac{33}{16}x^7 - \frac{429}{128}x^8 + \frac{715}{128}x^9 - \frac{2431}{256}x^{10}$$

where

$$G^2 = 1 + 2x - \frac{4199}{128}x^{11} + \text{ higher order terms.}$$

## Question 5

**(a)** If $G$ is given as in Question 3 then

$$F = G^2 \Rightarrow F - G^2 = 0$$

and letting $H = 1/G$ we get

$$F - (1/H)^2 = 0 \Rightarrow F - 1/H^2 = 0$$

as desired.

**(b)** We apply Newton's method to $P(H) = F - 1/H^2$ (so $P'(H) = 2/H^3$) to get the desired result. Let

$$H_{(i)} \equiv H \bmod x^{2^i} = H_0 + \cdots + H_{2^i-1}x^{2^i-1};$$

as $F - 1/H^2 \equiv 0 \bmod x$ we deduce that $H_0 = F_0 = 1$. The rest of the terms are given by the Newton scheme as follows;

$$\begin{aligned}
H_{(i+1)} &\equiv H_{(i)} - \frac{P(H_{(i)})}{P'(H_{(i)})} \bmod x^{2^{i+1}} \\
&\equiv H_{(i)} - \frac{F - 1/H_{(i)}^2}{2/H_{(i)}^3} \bmod x^{2^{i+1}} \\
&\equiv H_{(i)} - \frac{(FH_{(i)}^3 - H_{(i)})}{2} \bmod x^{2^{i+1}} \\
&\equiv \frac{2H_{(i)} - FH_{(i)}^3 + H_{(i)}}{2} \bmod x^{2^{i+1}} \\
&\equiv \frac{H_{(i)}(3 - FH_{(i)}^2)}{2} \bmod x^{2^{i+1}}.
\end{aligned}$$

which is the desired result. Note that informally we have that every iteration of the Newton scheme doubles the amount of correct terms, that is $H_{(i)} = H \bmod x^{2^i}$ which is proved in the next question.

**(c)** Assume that $H_{(i)} \equiv H \bmod x^{2^i}$. To prove that $H_{(i+1)} \equiv H \bmod x^{2^{i+1}}$ we will prove the equivalent statement $F - 1/H^2_{(i+1)} \equiv 0 \bmod x^{2^{i+1}}$ by showing

$$FH^2_{(i+1)} \equiv 1 \bmod x^{2^{i+1}} \tag{3}$$

Subbing in the identity from (b) into LHS (3) we get

$$\begin{aligned} FH^2_{(i+1)} &= F\left(\frac{H_{(i)}(3 - FH^2_{(i)})}{2} \bmod x^{2^{i+1}}\right)^2 \\ &\equiv \frac{FH^2_{(i)}\left(9 - 6FH^2_{(i)} + (FH^2_{(i)})^2\right)}{4} \bmod x^{2^{i+1}}. \end{aligned}$$

By our assumption we have that $F - 1/H^2_{(i)} \equiv 0 \bmod x^{2i}$ which implies that $FH^2_{(i)} \equiv 1 \bmod x^{2i}$. Therefore we can write $FH^2_{(i)}$ as $1 + \delta H$ where $\deg_x(\delta H) \geq 2^i$. Doing so and noting that $(\delta H)^2 \equiv 0 \bmod x^{2^{i+1}}$ we get

$$\begin{aligned} FH^2_{(i+1)} &\equiv \frac{(1+\delta H)(9 - 6(1+\delta H) + (1+\delta H)^2)}{4} \bmod x^{2^{i+1}} \\ &\equiv (1+\delta H)\left(1 - \delta H + \left(\frac{\delta H}{2}\right)^2\right) \bmod x^{2^{i+1}} \\ &\equiv (1+\delta H)(1-\delta H) \bmod x^{2^{i+1}} \\ &\equiv 1 - \delta H + \delta H - (\delta H)^2 \bmod x^{2^{i+1}} \\ &\equiv 1 \bmod x^{2^{i+1}} \end{aligned}$$

proving (3) which shows $H_{(i+1)} \equiv H \bmod x^{2^{i+1}}$.

**(d)** Let $T(n)$ denote the number of operations required to calculate $n$ terms of $H$ and recall that

$$H_0 + \cdots + H_{2^{n+1}-1}x^{2^{n+1}-1} = \frac{H_{(n)}(3 - FH^2_{(n)})}{2} \bmod x^{2^{n+1}}.$$

Therefore in order to calculate the first $2^{n+1}$ terms of $H$ requires that we know $H_{(n)}$ and do one multiplication in degree $2^n$ and three in degree $2^{n+1}$. This gives the recurrence:

$$\begin{aligned} T(2^{n+1}) = T(2\cdot 2^n) &\leq T(2^n) + \mathsf{M}(2^n) + 3\mathsf{M}(2^{n+1}) \qquad (4)\\ &\leq T(2^n) + 13\mathsf{M}(2^n) \qquad (5) \end{aligned}$$

(three multiplications in degree $2^{n+1}$ can be done with twelve multiplications in degree $2^n$ by naïve divide and conquer). By a Corollary from the lecture slides we have that (5) implies

$$T(2^n) \in O(\mathsf{M}(2^n)) \Rightarrow T(n) \in O(\mathsf{M}(n))$$

as desired.

**(e)** We have from the notes that calculating $1/H \bmod x^n$ costs $O(\mathsf{M}(n))$. As $G = 1/H$; inverting $H$ is equivalent to determining $G$. We need $n$ terms of $H$ to find the required inverse which also costs $O(\mathsf{M}(n))$ by **(d)** totalling $2O(\mathsf{M}(n))$ or $O(\mathsf{M}(n))$ as required.

## Question 6

time to complete $\approx 5$ hours