Theorem. $A(d) \asymp d$.

Theorem. $S(d) \asymp d$.

Theorem. *For every $\varepsilon > 0$, we have $M(d) \ll_{\varepsilon} d^{1+\varepsilon}$.*

Theorem. $M(d) \ll d\,(\log d) \log\log d$.

"Computational Complexity"

"Running Time"

Division

Problem: Given two positive integers n and m, determine the quotient q and the remainder r when n is divided by m. These should be integers satisfying

$$n = mq + r \quad \text{and} \quad 0 \le r < m.$$

Definition. Let $M'(d)$ denote an upper bound on the number of steps required to multiply two numbers with $\le d$ bits. Let $D'(d)$ denote an upper bound on the number of steps required to obtain q and r given n and m each have $\le d$ binary digits.

Theorem. *Suppose $M'(d)$ has the form $df(d)$ where $f(d)$ is an increasing function of d. Then $D'(d) \ll M'(d)$.*

Problem: Given two positive integers n and m, determine the quotient q and the remainder r when n is divided by m. These should be integers satisfying

$$n = mq + r \quad \text{and} \quad 0 \leq r < m.$$

We need only compute $1/m$ to sufficient accuracy.

Suppose n and m have $\leq s$ digits. If $1/m = 0.d_1d_2d_3d_4...$ (base 2) with $d_1, \ldots, d_s$ known, then

$$\frac{n}{m} = \frac{1}{2^s}(n \times d_1d_2 \ldots d_s) + \theta, \quad \text{where } 0 \leq \theta \leq 1.$$

Write this in the form

$$\frac{n}{m} = \frac{1}{2^s}(q'2^s + q'') + \theta,$$

so $n = mq' + \theta'$ where $0 \leq \theta' < 2m$. Try $q = q'$ and $q = q' + 1$.

Newton's Method

Say we want to compute $1/m$. Take a function $f(x)$ which has root $1/m$. If x' is an approximation to the root, then how can we get a better approximation? Take

$$f(x) = m - 1/x.$$

Starting with $x' = x_0$, this leads to the approximations

$$x_{n+1} = 2x_n - mx_n^2.$$

Note that if $x_n = (1 - \varepsilon)/m$, then $x_{n+1} = (1 - \varepsilon^2)/m$.

Algorithm from Knuth, Vol. 2, pp. 295-6

Algorithm R. Let v in binary be $v = (0.v_1v_2v_3\ldots)_2$, with $v_1 = 1$. The algorithm outputs z satisfying

$$|z - 1/v| \le 2^{-n}.$$

$z \in [0, 2]$

R1. [Initialize] Set $z \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$ and $k \leftarrow 0$.

R2. [Newton iteration] (At this point, $z \le 2$ has the binary form $(**.** \cdots *)_2$ with 2^k+1 places after the radix point.) Calculate z^2 exactly. Then calculate $V_k z^2$ exactly, where $V_k = (0.v_1v_2\ldots v_{2^{k+1}+3})_2$. Then set $z \leftarrow 2z - V_k z^2 + r$, where $0 \le r < 2^{-2^{k+1}-1}$ is added if needed to "round up" z so that it is a multiple of $2^{-2^{k+1}-1}$. Finally, set $k \leftarrow k+1$.

R3. [End Test] If $2^k < n$, go back to step R2; otherwise the algorithm terminates.

Algorithm from Knuth, Vol. 2, pp. 295-6

Algorithm R. Let v in binary be $v = (0.v_1v_2v_3\ldots)_2$, with $v_1 = 1$. The algorithm outputs z satisfying

$$|z - 1/v| \le 2^{-n}.$$

R1. [Initialize] Set $z \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$ and $k \leftarrow 0$.

R2. [Newton iteration] (At this point, $z \le 2$ has the binary form $(**.**\cdots*)_2$ with 2^k+1 places after the radix point.) Calculate z^2 exactly. Then calculate $V_k z^2$ exactly, where $V_k = (0.v_1v_2\ldots v_{2^{k+1}+3})_2$. Then set $z \leftarrow 2z - V_k z^2 + r$, where $0 \le r < 2^{-2^{k+1}-1}$ is added if needed to "round up" z so that it is a multiple of $2^{-2^{k+1}-1}$. Finally, set $k \leftarrow k+1$.

R3. [End Test] If $2^k < n$, go back to step R2; otherwise the algorithm terminates.

$k = 0$

$$(*) \qquad z_k \le 2 \quad \text{and} \quad |z_k - 1/v| \le 2^{-2^k}$$

Check me out!

$$\frac{1}{v} - z_{k+1} = v\left(\frac{1}{v} - z_k\right)^2 - z_k^2(v - V_k) - r$$

Algorith … $)_2$, with $v_1 = 1$. T …

$$|z - 1/v| \le 2 \ldots$$

R1. [Initialize] Set $z \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$ and $k \leftarrow 0$.

R2. [Newton iteration] (At this point, $z \le 2$ has the binary form $(**.** \cdots *)_2$ with 2^k+1 places after the radix point.) Calculate z^2 exactly. Then calculate $V_k z^2$ exactly, where $V_k = (0.v_1 v_2 \ldots v_{2^{k+1}+3})_2$. Then set $z \leftarrow 2z - V_k z^2 + r$, where $0 \le r < 2^{-2^{k+1}-1}$ is added if needed to "round up" z so that it is a multiple of $2^{-2^{k+1}-1}$. Finally, set $k \leftarrow k+1$.

R3. [End Test] If $2^k < n$, go back to step R2; otherwise the algorithm terminates.

$$(*) \qquad z_k \le 2 \quad \text{and} \quad |z_k - 1/v| \le 2^{-2^k}$$

Algorithm from Knuth, Vol. 2, pp. 295-6

Algorithm R. Let v in binary be $v = (0.v_1v_2v_3\ldots)_2$, with $v_1 = 1$. The algorithm outputs z satisfying

$$|z - 1/v| \le 2^{-n}.$$

R1. [Initialize] Set $z \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$ and $k \leftarrow 0$.

R2. [Newton iteration] (At this point, $z \le 2$ has the binary form $({*}{*}.{*}{*}\cdots{*})_2$ with $2^k{+}1$ places after the radix point.) Calculate z^2 exactly. Then calculate V_kz^2 exactly, where $V_k = (0.v_1v_2\ldots v_{2^{k+1}+3})_2$. Then set $z \leftarrow 2z - V_kz^2 + r$, where $0 \le r < 2^{-2^{k+1}-1}$ is added if needed to "round up" z so that it is a multiple of $2^{-2^{k+1}-1}$. Finally, set $k \leftarrow k+1$.

R3. [End Test] If $2^k < n$, go back to step R2; otherwise the algorithm terminates.

$$(*) \qquad z_k \le 2 \quad \text{and} \quad |z_k - 1/v| \le 2^{-2^k}$$

Division

Problem: Given two positive integers n and m, determine the quotient q and the remainder r when n is divided by m. These should be integers satisfying

$$n = mq + r \quad \text{and} \quad 0 \le r < m.$$

Definition. Let $M'(d)$ denote an upper bound on the number of steps required to multiply two numbers with $\le d$ bits. Let $D'(d)$ denote an upper bound on the number of steps required to obtain q and r given n and m each have $\le d$ binary digits.

Theorem. *Suppose $M'(d)$ has the form $df(d)$ where $f(d)$ is an increasing function of d. Then $D'(d) \ll M'(d)$.*

Algorithm from Knuth, Vol. 2, pp. 295-6

Algorithm R. Let v in binary be $v = (0.v_1v_2v_3\ldots)_2$, with $v_1 = 1$. The algorithm outputs z satisfying

$$|z - 1/v| \leq 2^{-n}.$$

R1. [Initialize] Set $z \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$ and $k \leftarrow 0$.

R2. [Newton iteration] (At this point, $z \leq 2$ has the binary form $(**.**\cdots*)_2$ with 2^k+1 places after the radix point.) Calculate z^2 exactly. Then calculate V_kz^2 exactly, where $V_k = (0.v_1v_2\ldots v_{2^{k+1}+3})_2$. Then set $z \leftarrow 2z - V_kz^2 + r$, where $0 \leq r < 2^{-2^{k+1}-1}$ is added if needed to "round up" z so that it is a multiple of $2^{-2^{k+1}-1}$. Finally, set $k \leftarrow k+1$.

R3. [End Test] If $2^k < n$, go back to step R2; otherwise the algorithm terminates.

Running Time:

$$2M'(4n) + 2M'(2n) + 2M'(n) + \cdots + O(n) \ll M'(n)$$

Division

Problem: Given two positive integers n and m, determine the quotient q and the remainder r when n is divided by m. These should be integers satisfying

$$n = mq + r \quad \text{and} \quad 0 \le r < m.$$

Definition. Let $M'(d)$ denote an upper bound on the number of steps required to multiply two numbers with $\le d$ bits. Let $D'(d)$ denote an upper bound on the number of steps required to obtain q and r given n and m each have $\le d$ binary digits.

Theorem. *Suppose $M'(d)$ has the form $df(d)$ where $f(d)$ is an increasing function of d. Then $D'(d) \ll M'(d)$.*

Elementary Number Theory

- **Modulo Arithmetic (definition, properties, & different notation)**