Definitions. Let $f(x)$ and $g(x)$ be functions with domain $[c, \infty)$ for some $c \in \mathbb{R}$ and range $\mathbb{R}$ and $\mathbb{R}^+$, respectively.

$f(x) = O(g(x))$ ("$f(x)$ is big-oh of $g(x)$")

$$\iff \exists\, C > 0, x_0 > 0 \text{ such that } |f(x)| \le Cg(x),\ \forall\, x \ge x_0$$

$f(x) \ll g(x)$ ("$f(x)$ is less than less than $g(x)$")

$$\iff f(x) = O(g(x))$$

$f(x) \gg g(x)$ ("$f(x)$ is greater than greater than $g(x)$")

$$\iff g(x) = O(f(x))$$

$f(x) \asymp g(x)$ ("the asymptotic order of $f(x)$ is $g(x)$")

$$\iff g(x) \ll f(x) \ll g(x) \text{ (or write } f(x) \gg\ll g(x))$$

$f(x) = o(g(x))$ ("$f(x)$ is little-oh of $g(x)$") $\iff \displaystyle\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$

$f(x) \sim g(x)$ ("$f(x)$ is aymptotic to $g(x)$") $\iff \displaystyle\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$

Definitions. Let $f(x)$ and $g(x)$ be functions with domain $[c, \infty)$ for some $c \in \mathbb{R}$ and range $\mathbb{R}$ and $\mathbb{R}^+$, respectively.

$f(x) = O(g(x))$ ("$f(x)$ is big-oh of $g(x)$")

$$\iff \exists\, C > 0, x_0 > 0 \text{ such that } |f(x)| \le C g(x), \;\forall\, x \ge x_0$$

$f(x) \ll g(x)$ ("$f(x)$ is less than less than $g(x)$")

$$\iff f(x) = O(g(x))$$

$f(x) \gg g(x)$ ("$f(x)$ is greater than greater than $g(x)$")

$$\iff g(x) = O(f(x))$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

Note: Analogous definitions exist if the domain is $\mathbb{Z}^+$.

Explicit Example: How quickly can we factor an $n \in \mathbb{Z}^+$?

We will want an "algorithm" that runs quickly (in a small number of steps) in comparison to the length of the input. One considers the length of the input n to be $\lfloor \log_2 n \rfloor + 1$ (corresponding to the number of bits n has). An algorithm runs in polynomial time if the number of steps (or bit operations) it takes is bounded above by a polynomial in the length of the input. An algorithm to factor n in polynomial time would require that it take $O\left((\log n)^k\right)$ steps (and that it factor n).

Addition and Subtraction

How fast do we add (or subtract) two numbers n and m?

How fast can we add (or subtract) two numbers n and m?

Definition. Let $A(d)$ denote the maximal number of steps required to add two numbers with $\leq d$ bits.

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

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

Multiplication

How fast do we multiply two numbers n and m?

How fast can we multiply two numbers n and m?

How many steps does it take to multiply a d bit number by 6?

How many steps does it take to divide a d bit number by 6?

(if it is divisible by 6)

$O(d)$ for these last two questions

Multiplication

How fast do we multiply two numbers n and m?

How fast can we multiply two numbers n and m?

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

Theorem. $M(d) \ll d^2$.

Can we do better? Yes

How can we see "easily" that something better is possible?

Attempt 1

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

- Suppose $M(d) \gg d^{1.5}$.
- Let d be large, and let $\varepsilon > 0$.
- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.
- From $nm = a_n a_m 2^{2r} + (a_n b_m + a_m b_n)2^r + b_n b_m$, deduce $M(d) \leq 4M(r+1) + O(r) \leq (4+\varepsilon)M(r+1)$.
- Hence, $M(d) \leq (4+\varepsilon)^s M\big((d + 2^{s+1} - 2)/2^s\big)$.
- Take $s = \lfloor \log_2 d \rfloor - C$ (with C big). Then $2^s \geq d/2^{C+1}$.
- Conclude, $M(d) \ll (4+\varepsilon)^{\log_2 d} = d^{\log(4+\varepsilon)/\log 2}$.

Attempt 2

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

- Suppose $M(d) \gg d^{1.5}$.
- Let d be large, and let $\varepsilon > 0$.
- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.
- From $nm =$????????????????????????????????????, deduce $M(d) \leq 3M(r+1) + O(r) \leq (3+\varepsilon)M(r+1)$.
- Hence, $M(d) \leq (3+\varepsilon)^s M\big((d + 2^{s+1} - 2)/2^s\big)$.
- Take $s = \lfloor \log_2 d \rfloor - C$ (with C big). Then $2^s \geq d/2^{C+1}$.
- Conclude, $M(d) \ll (3+\varepsilon)^{\log_2 d} = d^{\log(3+\varepsilon)/\log 2}$.

Attempt 2

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

- Suppose $M(d) \gg d^{1.5}$.
- Let d be large, and let $\varepsilon > 0$.
- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.
- From

  $$nm = a_n a_m 2^{2r} + \big((a_n + b_n)(a_m + b_m) - a_n a_m - b_n b_m\big) 2^r + b_n b_m,$$

  deduce $M(d) \leq 3M(r+2) + O(r) \leq (3+\varepsilon)M(r+2)$.
- Hence, $M(d) \leq (3+\varepsilon)^s M\big((d + 2^{s+1} - 2)/2^s\big)$.
- Take $s = \lfloor \log_2 d \rfloor - C$ (with C big). Then $2^s \geq d/2^{C+1}$.
- Conclude, $M(d) \ll (3+\varepsilon)^{\log_2 d} = d^{\log(3+\varepsilon)/\log 2}$.

Attempt 2

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

- Suppose $M(d) \gg d^{1.5}$.
- Let d be large, and let $\varepsilon > 0$.
- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.
- From

  $$nm = a_n a_m 2^{2r} + \big((a_n + b_n)(a_m + b_m) - a_n a_m - b_n b_m\big) 2^r + b_n b_m,$$

  deduce $M(d) \leq 3M(r+2) + O(r) \leq (3+\varepsilon) M(r+2)$.
- Hence, $M(d) \leq (3+\varepsilon)^s M\big((d + 2^{s+2} - 4)/2^s\big)$.
- Take $s = \lfloor \log_2 d \rfloor - C$ (with C big). Then $2^s \geq d/2^{C+1}$.
- Conclude, $M(d) \ll (3+\varepsilon)^{\log_2 d} = d^{\log(3+\varepsilon)/\log 2}$.

Attempt 2

Definition. Let $M(d)$ denote the number of steps required to multiply two numbers with $\leq d$ bits.

- Suppose $M(d) \gg d^{1.5}$.
- Let d be large, and let $\varepsilon > 0$.
- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.
- From

  $$nm = a_n a_m 2^{2r} + \big((a_n+b_n)(a_m+b_m) - a_n a_m - b_n b_m\big) 2^r + b_n b_m,$$

  deduce $M(d) \leq 3M(r+2) + O(r) \leq (3+\varepsilon)M(r+2)$.
- Hence, $M(d) \leq (3+\varepsilon)^s M\big((d + 2^{s+2} - 4)/2^s\big)$.
- Take $s = \lfloor \log_2 d \rfloor - C$ (with C big). Then $2^s \geq d/2^{C+1}$.
- Conclude, $M(d) \ll (3+\varepsilon)^{\log_2 d} = d^{\log(3+\varepsilon)/\log 2}$.

Theorem. $M(d) \ll d^2$.

- Conclude, $M(d) \ll (3+\varepsilon)^{\log_2 d} = d^{\log(3+\varepsilon)/\log 2}$.

$$\frac{\log 3}{\log 2} = 1.5849625$$

Theorem. $M(d) \ll d^{1.585}$.

HW: Due September 7 (Friday)
Page 3, Problems 1 and 2
Page 5, unnumbered homework (first set)
(you may use $(\log 5/\log 3)+\varepsilon$ instead of $\log 5/\log 3$)

Idea for Doing Better

- Let n and m have $\leq d$ bits, and write $n = a_n \times 2^r + b_n$ and $m = a_m \times 2^r + b_m$, where $r = \lfloor d/2 \rfloor$ and the a_j and b_j are integers with $b_j < 2^r$.

- From

$$nm = a_n a_m 2^{2r} + \big((a_n + b_n)(a_m + b_m) - a_n a_m - b_n b_m\big) 2^r + b_n b_m,$$

deduce $M(d) \leq 3M(r+2) + O(r) \leq (3+\varepsilon) M(r+2)$.

Think in terms of writing

$$n = a_n 2^{2r} + b_n 2^r + c_n \quad \text{and} \quad m = a_m 2^{2r} + b_m 2^r + c_m,$$

where $r = \lfloor d/3 \rfloor$.

How many multiplications does it take to expand nm?

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

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

Theorem. *Given distinct numbers* $x_0, x_1, \ldots, x_k$ *and numbers* $y_0, y_1, \ldots, y_k$, *there is a unique polynomial* f *of degree* $\le k$ *such that* $f(x_j) = y_j$ *for all* j.

Lagrange Interpolation:

$$f(x) = \sum_{i=0}^{k} \left(\prod_{\substack{0 \le j \le k \\ j \ne i}} \frac{x - x_j}{x_i - x_j} \right) y_i$$

Theorem. *Given distinct numbers $x_0, x_1, \ldots, x_k$ and numbers $y_0, y_1, \ldots, y_k$, there is a unique polynomial f of degree $\leq k$ such that $f(x_j) = y_j$ for all j.*

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^k \\ 1 & x_1 & x_1^2 & \ldots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \ldots & x_k^k \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix}$$

$$\det \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^k \\ 1 & x_1 & x_1^2 & \ldots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \ldots & x_k^k \end{pmatrix} = \prod_{0 \leq i < j \leq k} (x_j - x_i)$$