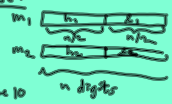
CSE 750

Two algos for multiplication of integers:

Divide-and-conquer

1st:



base 10

$$m_1 = h_1 \cdot 10^{n/2} + l_1$$
$$m_2 = h_2 \cdot 10^{n/2} + l_2$$

base 2:

$$m_1 = h_1 \cdot 2^{n/2} + l_1$$
$$m_2 = h_2 \cdot 2^{n/2} + l_2$$

$$m_1 m_2 = (h_1 \cdot 2^{n/2} + l_1)(h_2 \cdot 2^{n/2} + l_2)$$

$$= \underbrace{h_1 h_2}_{\text{rec call}} 2^{n} + (\underbrace{l_1 h_2}_{\text{rec call}} + \underbrace{h_1 l_2}_{\text{rec call}}) 2^{n/2} + \underbrace{l_1 l_2}_{\text{rec call}}$$

arithmetic shift

$T(n)$ = time to mult two n-bit integers

$$T(n) = 4T(\tfrac{n}{2}) + \overbrace{cn}^{\text{everything else}}$$

(c is some constant)

$T(1) = c$

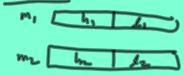
$T(2) = 4T(1) + 2c = 6c$

$T(4) = 4T(2) + 4c = 28c$

$\vdots$

$T(n) \approx c' n^2$ (c' some constant)

Algo 2:



$$(2^{n/2} h_1 + l_1)(2^{n/2} h_2 + l_2)$$
$$= h_1 h_2 2^n + (l_1 h_2 + h_1 l_2) 2^{n/2} + l_1 l_2$$

$$r := \underbrace{(\underbrace{h_1 \oplus l_1}_{n/2})(\underbrace{h_2 \oplus l_2}_{n/2})}_{\text{rec call}}$$

$$r := h_1 h_2 + l_1 h_2 + h_1 l_2 + l_1 l_2$$

compute r

then

$$m_1 m_2 = \underbrace{h_1 h_2}_{\text{rec call}} 2^n \oplus (r \ominus h_1 h_2 \ominus l_1 l_2) 2^{n/2} \oplus \underbrace{l_1 l_2}_{\text{rec call}}$$

$$T(n) = 3T(\tfrac{n}{2}) + dn$$

$d > c$

---

code 1:

```
sum = 0;
for(i=1; i<n; i *=2)
  for(j=0; j<i; j++)
    sum++;
```

proportional to n

code 2:

```
sum = 0;
for(i=1; i<n; i*=2)
  for(j=i; j<n; j++)
    sum++;
```

proportional to n lg n.

Time for code 1:

$$1 + 2 + 4 + 8 \cdots 2^k = 2^{k+1} - 1 < 2n$$

$$2^k < n \le 2^{k+1}$$

Time for code 2:

$$(n-1) + (n-2) + (n-4) + \cdots + (n - 2^k)$$

$$= kn - (\underbrace{1 + 2 + 4 + \cdots + 2^k}_{<2n})$$

$$> \underbrace{(k-2)n}_{\text{code 2}} \quad \text{vs} \quad \underbrace{2n}_{\text{code 1}}$$

$$2^k < n \le 2^{k+1}$$

apply $\log_2$ to all sides:

$$\log_2 2^k < \log_2 n \le \log_2 2^{k+1}$$
$$k < \log_2 n \le k+1$$
$$k \doteq \log n$$
$$k-2 \doteq \log n$$

$\lg = \log_2 \qquad \lg 2^n = n$

Aug 23-12:46 PM

Asymptotic analysis

1. don't care about small inputs
2. don't care about constant factors

$100n < n^2$

Let $f(n)$ and $g(n)$ be two functions of n (assume $n \geq 0$ and an integer)

f, g are eventually positive

(there exists N such that, for all $n \geq N$, $f(n) > 0$, $g(n) > 0$.)

Def: We say that

$f(n) = O(g(n))$

if there exist constant $C > 0$ and integer $N \geq 0$ such that,

for all $n \geq N$

$$f(n) \leq C \cdot g(n).$$

"f grows no faster than g (asymptotically)"

Aug 23-1:34 PM