LECTURE 2

Outline:

1) Integer Multiplication

→ KARATSUBA'S ALGORITHM

2) Solving Recurrence Relations.

(B1a) INTEGERS

= Numbers stored as an array of digits

Why: * Architectures support 64 bit integers natively, larger integers implemented in software.

* Application: Cryptography .... .

INTEGER ADDITION

INPUT: $x[1..n]$, $y[1..n]$ n-digit numbers

GOAL: $z = x + y$ - an $(n+1)$ digit number

ALGORITHM:

$$
\begin{array}{rcccccc}
 & & 1 & & & & \\
x = & 1 & 7 & 9 & 2 & 1 & 3 \\
 & & & & & & + \\
y = & 2 & 1 & 9 & 6 & 7 & 1 \\
\hline
 & 3 & 9 & 8 & 8 & 8 & 4
\end{array}
$$

$\longleftarrow$

$\longleftarrow$ n+1 digits $\longrightarrow$

RUNTIME: Each digit takes $O(1)$ time

In total $O(n)$ time

INTEGER MULTIPLICATION

INPUT: $a[1..n]$, $b[1..n]$ n digit numbers

OUTPUT: $c[1..2n] = a * b$ $2n$ digit number.

GRADE SCHOOL ALGO

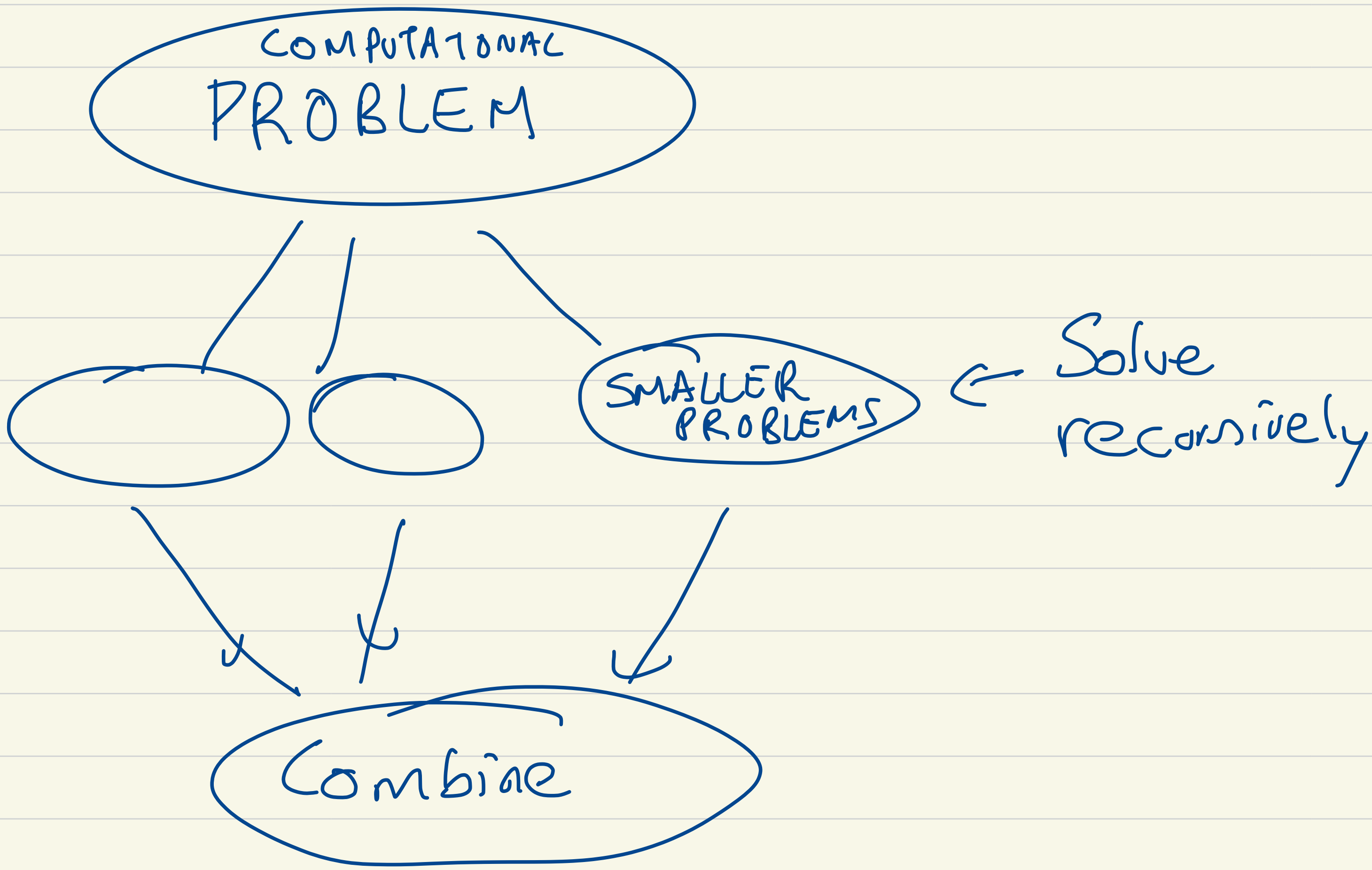
$$
\begin{array}{r}
1234 \; * \; 2121 \\
\hline
1234 \\
2468\,* \\
1234\,*\,* \\
2468\,*\,*\,* \\
\hline
2617314 \\
\hline
\end{array}
$$

n rows

RUNTIME: Adding n-digit numbers, n times

$$\Rightarrow \Theta(n * n) = \Theta(n^2)$$

DIVIDE AND CONQUER



INTEGER MULTIPLICATION

a_L	a_R

b_L	b_R

$$a = (10)^{n/2} a_L + a_R \qquad b = 10^{n/2} b_L + b_R$$

$$123\,456 \qquad\qquad 654\,321$$

$$= (123) \cdot 10^3 + 456 \qquad\qquad = (654) \cdot 10^3 + 321$$

$$a * b = \left(10^{n/2} a_L + a_R\right)\left(10^{n/2} b_L + b_R\right)$$

$$= (10^n) \cdot \underbrace{a_L b_L} + 10^{n/2}\left[\underbrace{a_L b_R} + \underbrace{a_R b_L}\right] + \underbrace{a_R b_R}$$

products of $n/2$ bit integers

MULT$(a[1..n], b[1..n])$

* IF $n<2$ return $a[1]*b[1]$

* SPLIT $a \to a_L, a_R$
  $b \to b_L, b_R$

* $P1 \leftarrow$ MULT(a_L, b_L)
  $P2 \leftarrow$ MULT(a_L, b_R)
  $P3 \leftarrow$ MULT(a_R, b_R)
  $P4 \leftarrow$ MULT(a_R, b_L)

* RETURN $10^n \cdot P1 + 10^{n/2} \cdot [P2+P3] + P4$

Append n zeroes

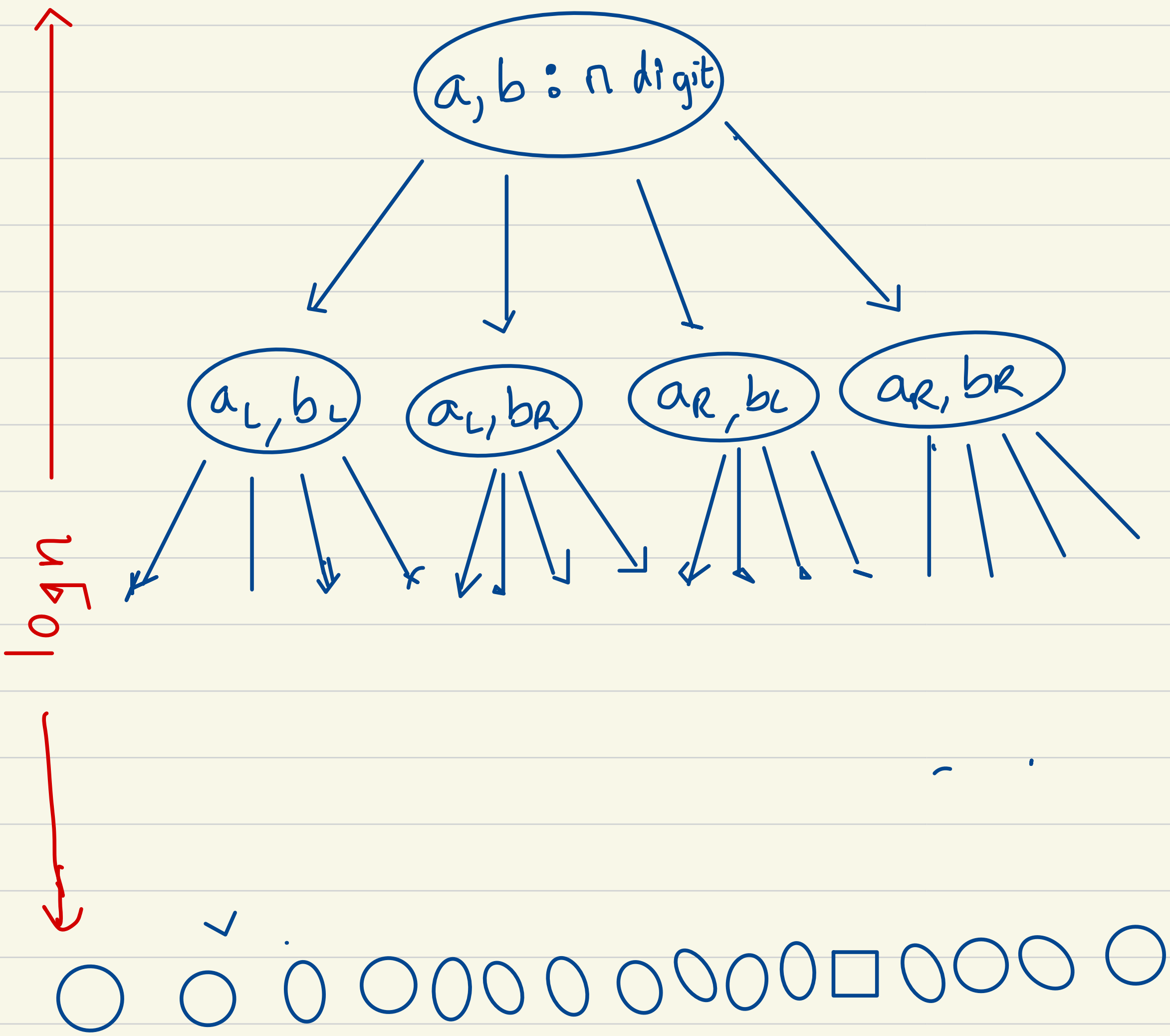
Append n/2 zeros

RUNTIME:

$T[n]$ = time taken by MULT on n-digit numbers

$$T[n] = 4 \cdot T[n/2] + c \cdot n$$

for some constant c.



#NODES	WORK PER NODE
1	$c \cdot n$
4	$c \cdot \left(n/2\right)$
4^2	$c \cdot \left(n/2^2\right)$
4^k	$c \cdot \left(\frac{n}{2^k}\right)$
$4^{\log n}$	$c \cdot \left(\frac{n}{2^{\log n}}\right)$

$$= 1 \cdot cn + 4\left(\frac{cn}{2}\right) + 4^2 \cdot \left(\frac{cn}{2^2}\right) + \cdots + 4^{\log n} \cdot \left(\frac{cn}{2^{\log n}}\right)$$

$$= \Theta\left(\frac{4^{\log n}}{2^{\log n}} \cdot cn\right) = \Theta\left(n^2\right)$$

FACTS:

1) Sum of A Geometric Progression with ratio > 1

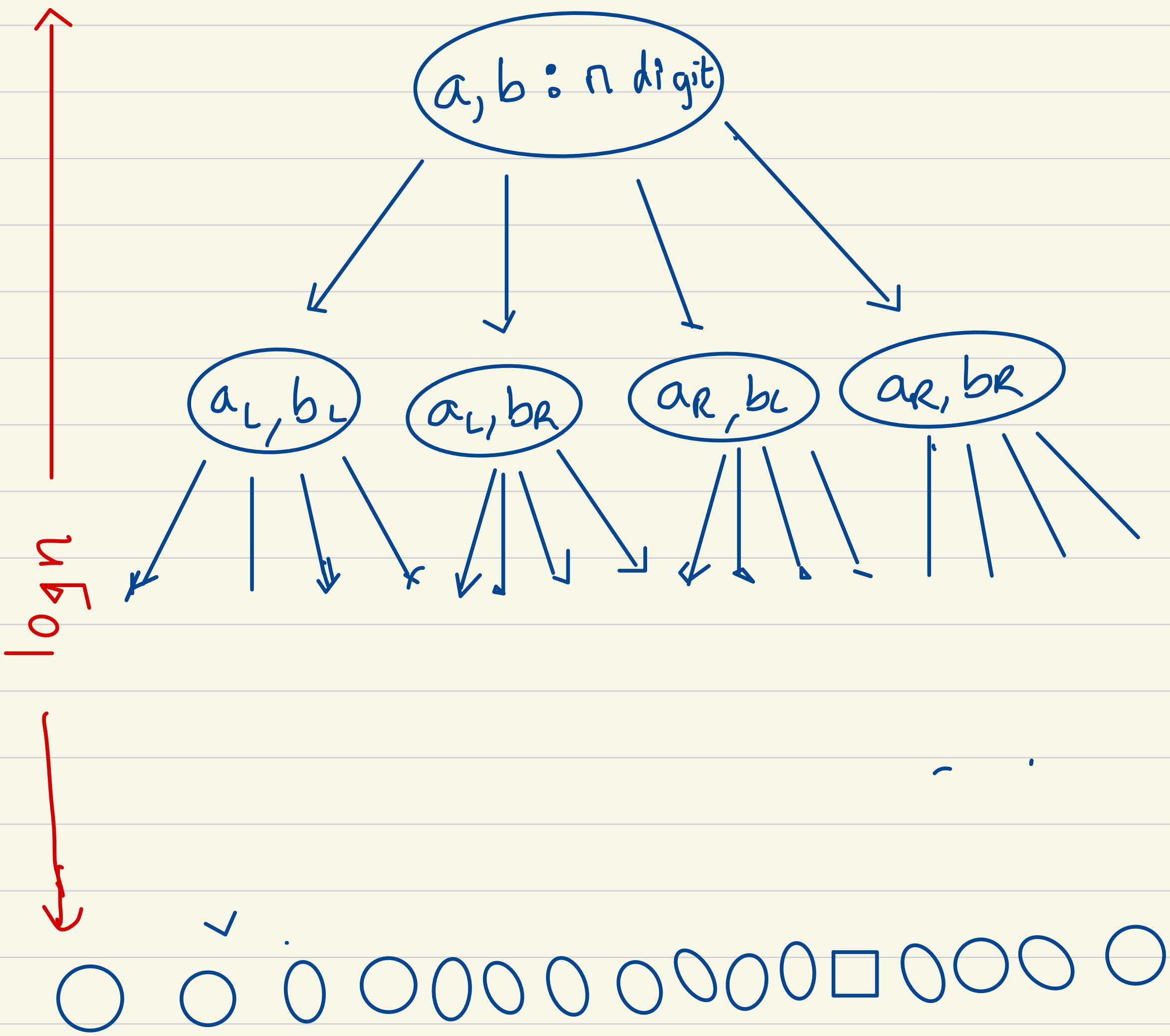
$$= O(\text{last term})$$

2) $$2^{\log_2 n} = n$$

$$4^{\log_2 n} = (2^2)^{\log n} = (2^{\log n})^2 = n^2$$

3) $\forall a, b \quad a^{\log b} = b^{\log a}$

IDEA:



# NODES	WORK PER NODE
1	$c \cdot n$
4	$c \cdot (n/2)$
4^2	$c \cdot (n/2^2)$
4^k	$c \cdot \left(\frac{n}{2^k}\right)$
$4^{\log n}$	$c \cdot \left(\frac{n}{2^{\log n}}\right)$

$$= 1 \cdot cn + 4\left(\frac{cn}{2}\right) + 4^2 \cdot \left(\frac{cn}{2^2}\right) + \dots + 4^{\log n} \cdot \left(\frac{cn}{2^{\log n}}\right)$$

$$= \Theta\left(\frac{4^{\log n}}{2^{\log n}} \cdot cn\right) = \Theta(n^2)$$

GOAL: Implement an n digit multiplication using 3 $n/2$ digit multiplications.

$$a = 10^{n/2} \cdot a_L + a_R$$

$$b = 10^{n/2} \cdot b_L + b_R$$

$$a \cdot b = 10^{n} \cdot a_L \cdot b_L + 10^{n/2} [a_L b_R + a_R b_L] + a_R b_R$$

$$= 10^{n} \underbrace{a_L \cdot b_L}_{P1} + 10^{n/2} [\underbrace{(a_L + a_R)(b_L + b_R)}_{P3} - \underbrace{a_L b_L}_{P1} - \underbrace{a_R b_R}_{P2}] + \underbrace{a_R b_R}_{P2}$$

KARATSUBA'S ALGORITHM

MULT(a[1..n], b[1..n])

* IF $n < 2$ RETURN $a[1] \cdot b[1]$

* SPLIT $a \to a_L, a_R$
  $b \to b_L, b_R$

* $P1 \leftarrow$ MULT(a_L, b_L)
  $P2 \leftarrow$ MULT(a_R, b_R)
  $P3 \leftarrow$ MULT$(a_L + a_R, b_L + b_R)$

O(n) time additions

* RETURN $10^n \cdot P1 + 10^{n/2}[P3 - P1 - P2] + P2$

$$T[n] = 3\,T[n/2] + c \cdot n$$ for some const c

Solving Recurrences

Example 1: $T(n) = T(n-1) + \sqrt{n}$

(unroll the recursion)

$$
\begin{aligned}
T(n) &= T(n-1) + \sqrt{n} \\
&= T(n-2) + \sqrt{n-1} + \sqrt{n} \\
&= T(1) + \sqrt{2} + \sqrt{3} + \dots + \sqrt{n}
\end{aligned}
$$

* Each term $< \sqrt{n}$ $\Rightarrow$ $T(n) < n \cdot \sqrt{n} = n^{1.5}$

* Look at last $n/2$ terms

$$T(n) \geq \sqrt{n/2} + \sqrt{\frac{n}{2}+1} \dots + \sqrt{n}$$

$\geq \frac{n}{2}$ terms each at least $\sqrt{\frac{n}{2}}$

$$\geq \frac{n}{2} \cdot \sqrt{\frac{n}{2}} = \frac{n^{1.5}}{2\sqrt{2}}$$

So

$$n^{1.5} \geq T(n) \geq n^{1.5}/2\sqrt{2}$$

$$T[n] = 2\,T(n/3) + n$$

$$= 2\left[2T(n/9) + n/3\right] + n$$

$$= 2^2\,T(n/3^2) + \frac{2n}{3} + n$$

$$= 2^2\left[2T\left(n/3^3\right) + \frac{n}{3^2}\right] + \frac{2n}{3} + n$$

$$= 2^3\, T\left(n/3^3\right) + \frac{2^2}{3^2}n + \frac{2n}{3} + n$$

$$= 2^k\, T\left(n/3^k\right) + \underbrace{\left[\left(\frac{2^{k-1}}{3^{k-1}}\right)\cdot n + \dots + \left(\frac{2}{3}\right)n + n\right]}_{\Theta(\text{last term}) = O(n)}$$

$$k = \log_3 n \iff 3^k = n$$

$$= 2^{\log_3 n} + \Theta(n)$$

$$= n^{\log_3 2} + \Theta(n) = \Theta(n)$$

MASTER THEOREM:

Suppose function $T: \mathbb{N} \to \mathbb{R}^+$ satisfies

$$T[n] = a\,T\left(\frac{n}{b}\right) + O(n^c) \quad \text{then}$$

where $b > 1$ $a, c > 0$ are constants

CASE 1: $c < \log_b a$ $\quad T(n) = O(n^{\log_b a})$

CASE 2: $c = \log_b a$ $\quad T[n] = O(n^c \log n)$

CASE 3: $c > \log_b a$ $\quad T[n] = O(n^c)$