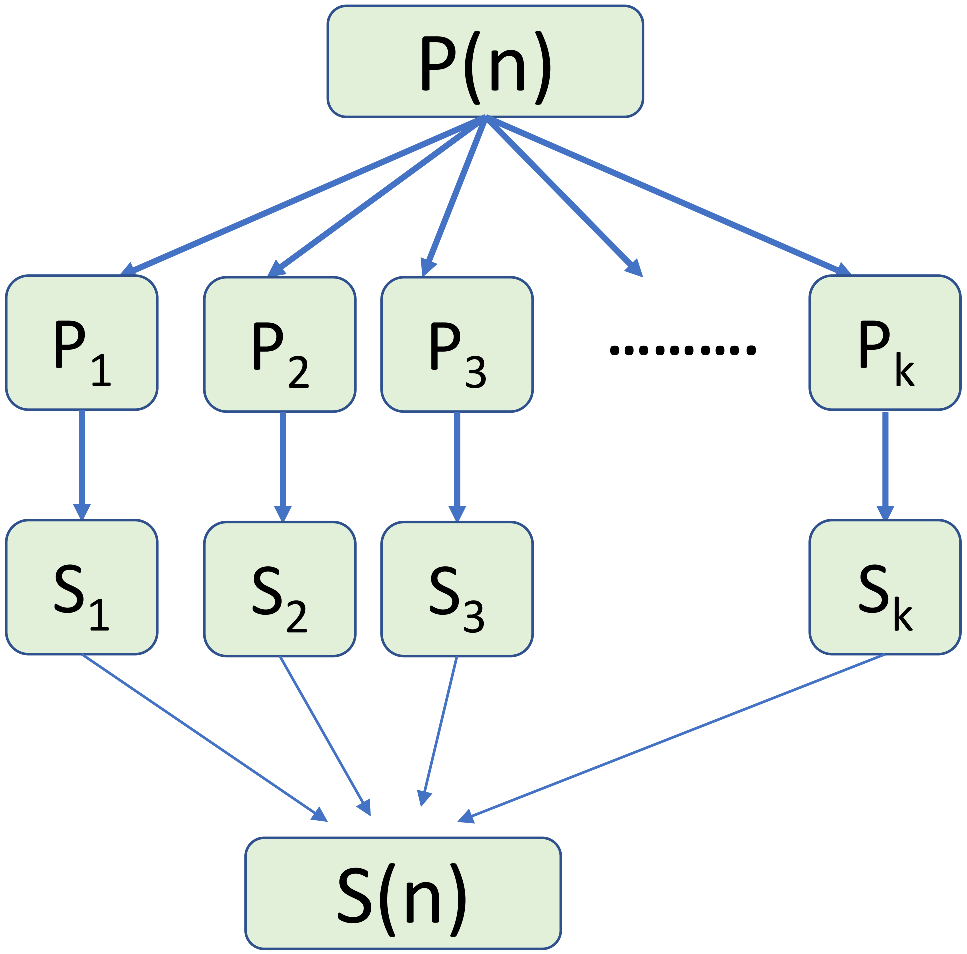
CIS 675
Recurrences

Dr. Asif Salekin

Divide & Conquer



Recursive in nature

- **Smaller sub-problems will be same as the problem P(n)**
  You can not transform them into another problem.

Example:

If P(n) is sorting an array of size n. The sub-problems can only be sorting an array of size m, where $m<n$.

- **Need a strategy to combine the solutions of the subproblems.**

Warmup!

$$(1+a+a^2+\ldots\ldots.+a^L)(a-1) = a+a^2+a^3+\ldots\ldots +a^L +a^{L+1}$$
$$-1-a-a^2-a^3- \ldots\ldots -a^L$$
$$= a^{L+1}-1$$

$$(1+a+a^2+\ldots\ldots.+a^L) = \frac{a^{L+1}-1}{a-1}$$

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1}-1}{a-1}$$

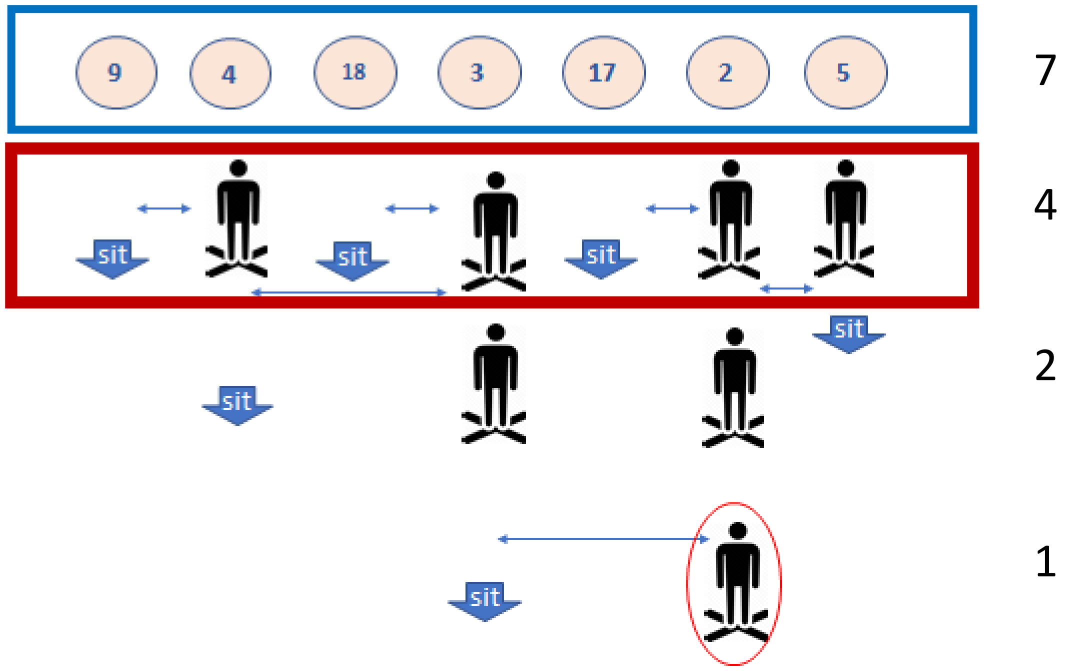
Who has smallest SU ID in the first row?

When we can not memorize any number!

1. Stand
2. Greet a neighbor (stop if you are the only person standing) (do not communicate with more than one!)
3. If you have larger SU ID sit
   If you have smaller SU ID compared to your neighbor's, remain standing
4. If you are standing & you have neighbor, go to 2

How fast does it work?

	1	2	3	4
	Stand	Greet	Sit/stay	repeat
T(n)=		1	1	T ($\lceil n/2 \rceil$)



T(n) is a function: steps to finish in a room of n students

$T(n) = 1 + 1 + T(\lceil n/2 \rceil)$

& $T(1) = 3$

How can we solve it?

Recurrence?

$T(n) = 2 + T(\lceil n/2 \rceil)$
& $T(1)=3$

- **We are not going for exact solution!**
- **We will find an asymptotic bound!**

Solve a simpler case when n is power of 2!

$T(n) = 2 + T(\lceil n/2 \rceil)$
& $T(1)=3$

$$T(2^K) = 2 + T(2^{K-1})$$
$$+ 2 + T(2^{K-2})$$
$$+ 2 + T(2^{K-3})$$
$$+ 2 + T(2^{K-4})$$
$$+2 + T(1) \Rightarrow 3$$

How many 2's involve?

$2^{K-K} = 2^0 = 1$

$$T(2^K) = 2K+3 = 2\log_2(2^K)+3$$

Solve a simpler case when n is power of 2!

$T(n) = 2 + T(\lceil n/2 \rceil)$
$\& \; T(1) = 3$

$$
\begin{aligned}
T(2^K) &= 2 + T(2^{K-1}) \\
&= 2 + 2 + T(2^{K-2}) \\
&= 2 + 2 + 2 + T(2^{K-3}) \\
&= 2 + \overbrace{2 + \ldots + 2}^{K-1} + T(2^0) \\
&= 2K + 3
\end{aligned}
$$

$\forall\, 0 < n < m,\; T(n) \leq T(m)$

$$T(m) \leq T(2^{\lceil \log(m) \rceil}) = 2\lceil \log(m) \rceil + 3$$

Asymptotic notation

Set of functions

$f(X)=O(g(X))$ at most within cost of g   for large n

$$\left\{ \begin{array}{l} \textbf{function f: there exist positive constants c, } n_0 \textbf{ such that} \\ \quad\quad \textbf{for all } n>n_0,\ 0 \leq f(n) \leq c \times g(n) \end{array} \right\}$$

Important note:
O(g(X)) is actually a set!

When we say "f(x)=O(g(x)), we are actually
Saying that f(X) ∊ O(g(x))

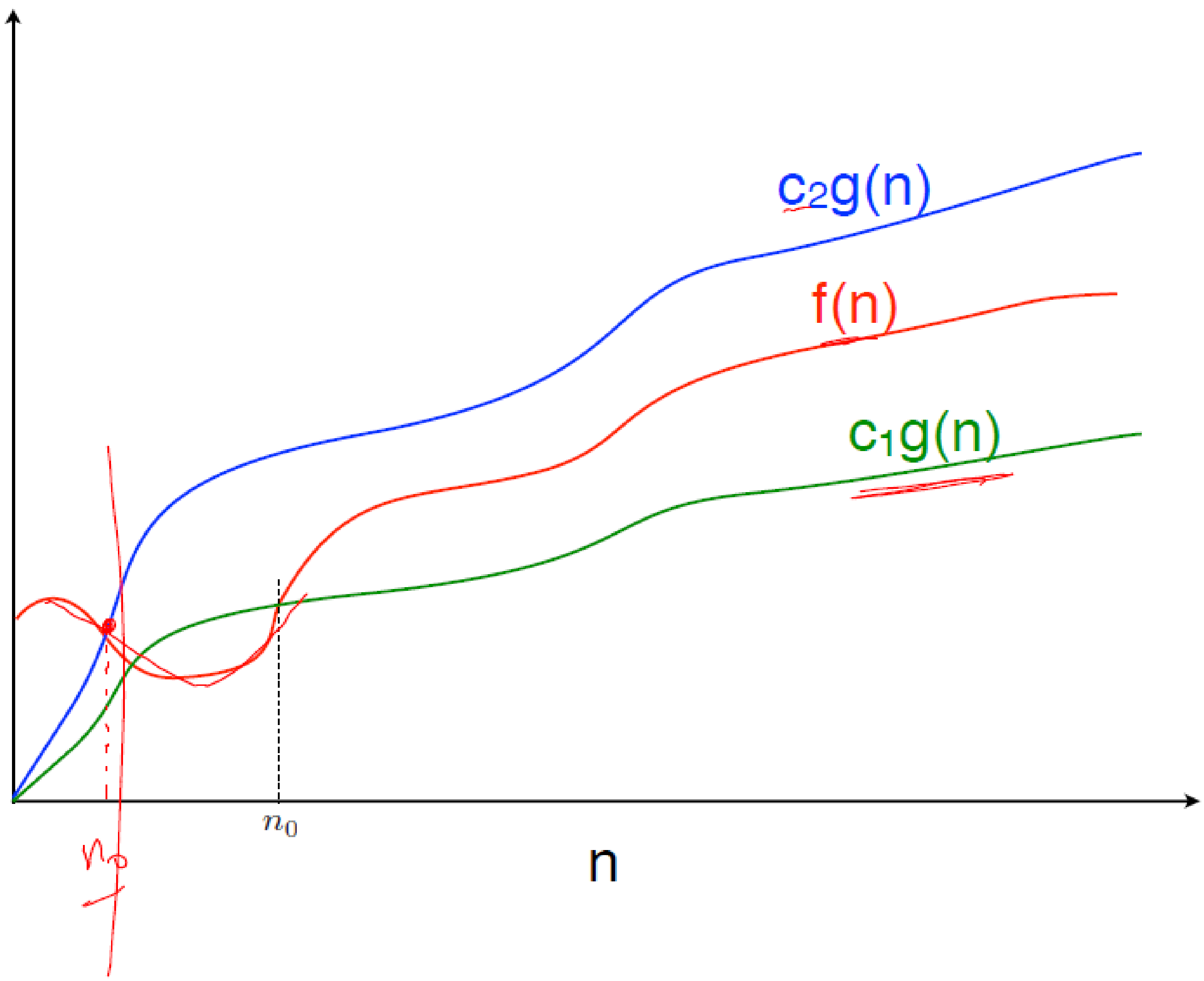
Asymptotic notation

O(g) at most within const of g for large n

Ω(g) at least within const of g for large n

Θ(g) within const of g for large n

Asymptotic notation



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

$$f(n) = \Theta(g(n))$$

$$f(n) = \Omega(g(n))$$

Asymptotic notation

$\Theta(g)$ ***For all of our algorithms we want to proof the theta bound***

$\Theta(g(n))$ = {f(n): there exist positive constants c_1, c_2 and n_0

such that $0 \leq c_1 \times g(n) \leq f(n) \leq c_2 \times g(n)$ for all $n \geq n_0$}

The above definition means, if f(n) is theta of g(n), then the value f(n) is always between $c_1 \times g(n)$ and $c_2 \times g(n)$ for large values of n ($n \geq n_0$).

Asymptotic notation

$$T(m) \leq T(2^{\lceil \log(m) \rceil}) = 2\lceil \log(m) \rceil + 3$$

Upper bound

$T(m)=O(\log m)$

$\Omega(m)$?????

Main Ideas

- Break large problem into smaller ones.
- Use recurrence relation to analyze the running time.
- We use asymptotic notation to simplify the analysis.

How to solve recurrence relations?

- Tree method
- Guess & check method (induction)
- Cookbook method “Master Theorem”
- Substitution Technique

Multiplication

				1	7	8	9
			×	1	4	3	2
				3	5	7	8
			5	3	6	7	
		7	1	5	6		
1	7	8	9				

N distinct numbers

$$\underbrace{\text{4 multiplication} \quad \text{4 addition}}_{\text{2n operations}}$$

4 multiplication **4 addition**

4 multiplication **4 addition**

4 multiplication **4 addition**

4 multiplication **4 addition**

n rows

$O(n^2)$

Main Ideas

- Break large problem into smaller ones.
- Use recurrence relation to analyze the running time.
- We use asymptotic notation to simplify the analysis.

Multiplication

$17 \times 100 + 89 \qquad 14 \times 100 + 32$

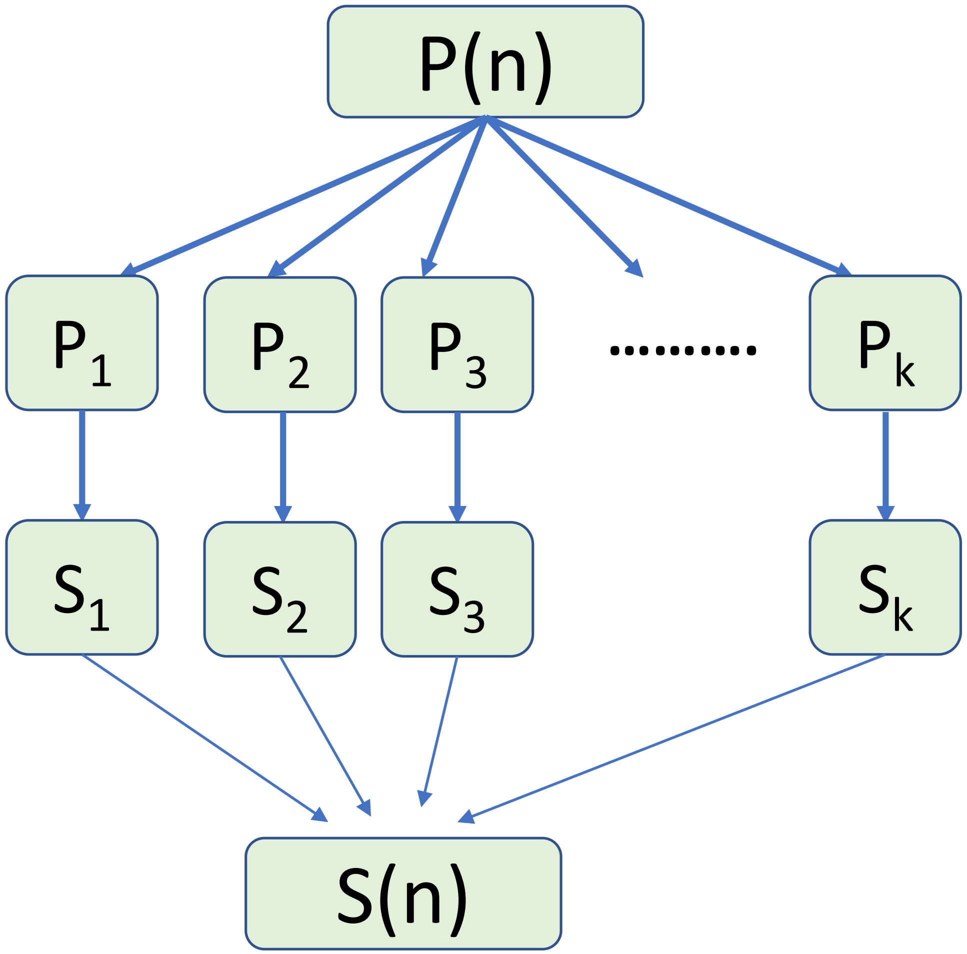
$\underbrace{\boxed{1}\ \boxed{7}}_{a}\ \underbrace{\boxed{8}\ \boxed{9}}_{b} \times \underbrace{\boxed{1}\ \boxed{4}}_{c}\ \underbrace{\boxed{3}\ \boxed{2}}_{d}$

n digit

a,b,c,d are $\frac{n}{2}$ digit

$(a\times c)(100^2) + (a\times d + b\times c)\ (100) + b\times d$

Divide & Conquer



Recursive in nature

- **Smaller sub-problems will be same as the problem P(n)**
  You can not transform them into another problem.

Example:

If P(n) is sorting an array of size n. The sub-problems can only be sorting an array of size m, where m<n.

- **Need a strategy to combine the solutions of the subproblems.**

Multiplication

$17 \times 100 + 89$ $\quad$ $14 \times 100 + 32$

$$\underbrace{1\ 7}_{a}\ \underbrace{8\ 9}_{b} \times \underbrace{1\ 4}_{c}\ \underbrace{3\ 2}_{d}$$

n digit

a,b,c,d are $\frac{n}{2}$ digit

$$(a \times c)(100^2) + (a \times d + b \times c)(100) + b \times d$$

Let's analyze how well it works!

Multiplication

Mult(ab,cd)

1	7	8	9	×	1	4	3	2

$\underbrace{17}_{a}\ \underbrace{89}_{b} \times \underbrace{14}_{c}\ \underbrace{32}_{d}$

$ac\times100^2+(ad+bc)\times100+bd$

BASE CASE:

return b × d if inputs are 1 digit

ELSE:

Compute X=mult(a,c) → $T(\frac{n}{2})$

Compute Y=mult(a,d) → $T(\frac{n}{2})$

Compute Z=mult(b,c) → $T(\frac{n}{2})$

Compute W=mult(b,d) → $T(\frac{n}{2})$

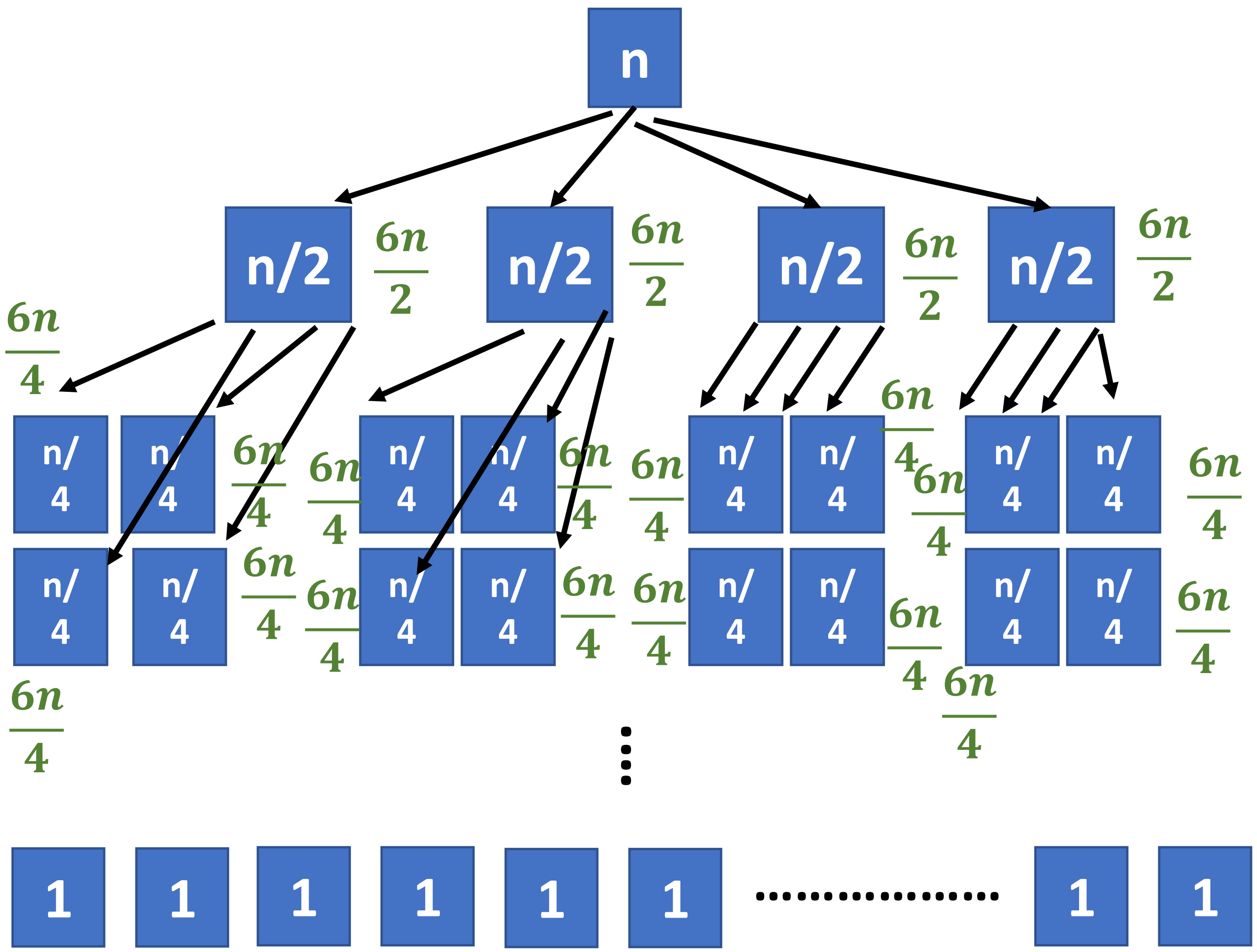
$4\times T(\frac{n}{2})$

Return $X \times ((10)^{(\text{number of digit of a})})^2 + (Y+Z)\times(10)^{(\text{number of digit of a})} + W$

7n steps

$T(n)=4T(\frac{n}{2}) + 6n$

$T(n)=4T(\frac{n}{2}) + 6n$, base case: $T(1)=1$



0	$6n$
1	$4\times\frac{6n}{2} = 2^1\times 6n$
2	$16\times\frac{6n}{4} = 2^2\times 6n$
3	$64\times\frac{6n}{8} = 2^3\times 6n$
$\lceil \log n \rceil$	$2^{\lceil \log n \rceil}\times 6n$

Calculations:

$$(1+a+a^2+\ldots\ldots+a^L) = \frac{a^{L+1}-1}{a-1}$$

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1}-1}{a-1}$$

$$T(n) = 3n + 3n\times 2 + 3n\times 2^2 + \ldots\ldots\ldots 3n\times 2^{\lceil \log n \rceil}$$

$$= 3n \times (1 + 2 + 2^2 + \ldots\ldots\ldots + 2^{\lceil \log n \rceil})$$

n

$$= \mathbf{3n} \times \left[\frac{2^{1+\lceil \log n \rceil}-1}{2-1} \right] = 3n \times [2 \times 2^{\lceil \log n \rceil} - \mathbf{1}]$$

$$= 3n[2n-1]$$

$$= 6n^2 - 3n$$

$$= O(n^2)$$

Karatsuba

$$\underbrace{\boxed{1}\ \boxed{7}}_{a}\ \underbrace{\boxed{8}\ \boxed{9}}_{b} \times \underbrace{\boxed{1}\ \boxed{4}}_{c}\ \underbrace{\boxed{3}\ \boxed{2}}_{d}$$

$$(a\times c)(100^2) + (a\times d + b\times c)(100) + b\times d$$

$$(a+b)(c+d)=ac+ad+bc+bd$$

$$ad+bc=(a+b)(c+d)-ac-bd$$

$(a \times c)(100^2) + (a \times d + b \times c)(100) + b \times d$

Karatsuba

$$\underbrace{\boxed{1}\ \boxed{7}}_{\mathbf{a}\ \ n/2}\ \underbrace{\boxed{8}\ \boxed{9}}_{\mathbf{b}\ \ n/2} \times \underbrace{\boxed{1}\ \boxed{4}}_{\mathbf{c}\ \ n/2}\ \underbrace{\boxed{3}\ \boxed{2}}_{\mathbf{d}\ \ n/2}$$

Recursively compute:

1. ac,bd,(a+b)(c+d)
2. ad+bc=(a+b)(c+d) –ac-bd
3. $ac \times 100^2 + (ad+bc) \times 100 + bd$

$3T(\frac{n}{2})$

2 addition,

4n subtraction

4n addition

Approximately,
Not exactly

Karatsuba (ab,cd)

1	7	8	9	×	1	4	3	2

$\underbrace{17}_{a}\ \underbrace{89}_{b} \times \underbrace{14}_{c}\ \underbrace{32}_{d}$

BASE CASE:

return b × d if inputs are 1 digit

ELSE:

Compute ac=karatsuba(a,c) → T($\frac{n}{2}$)

Compute bd=karatsuba(b,d) → T($\frac{n}{2}$)

Compute t=karatsuba((a+b),(c+d)) → T($\frac{n}{2}$) $+ 2n$

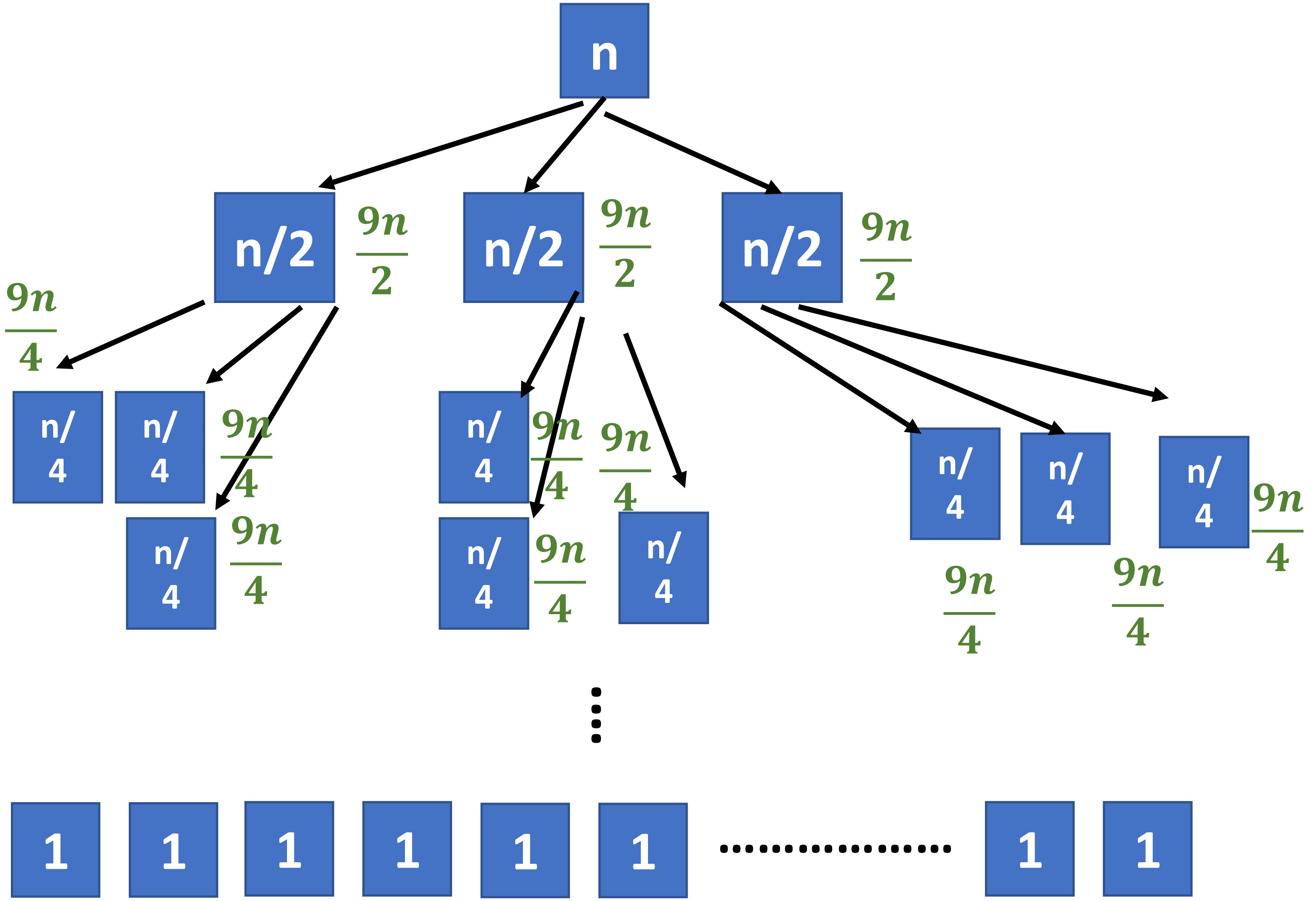
mid=t-ac-bd → **3n**

T(n)=3T($\frac{n}{2}$) +9n

Ignoring issue of carries

Return ac × ((10)$^{\text{(number of digit of a)}}$)^2 +mid×(10)$^{\text{(number of digit of a)}}$ +bd } **4n steps**

$T(n)=3T(\frac{n}{2})+9n$



0 $9n$

1 $3\times\frac{9n}{2} = \left(\frac{3}{2}\right)^{1}\times 9n$

2 $9\times\frac{9n}{4} = \left(\frac{3}{2}\right)^{2}\times 9n$

3 $27\times\frac{9n}{8} = \left(\frac{3}{2}\right)^{3}\times 9n$

$\lceil \log n \rceil$ $\left(\frac{3}{2}\right)^{\lceil \log n \rceil}\times 9n$

Calculations:

$$(1+a+a^2+\ldots\ldots+a^L) = \frac{a^{L+1}-1}{a-1}$$

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1}-1}{a-1}$$

$$T(n) = 9n + \frac{3}{2} \times 9n + \left(\frac{3}{2}\right)^2 \times 9n + \ldots\ldots + \left(\frac{3}{2}\right)^{\lceil \log n \rceil} \times 9n$$

$$= 9n \times \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2 + \ldots\ldots + \left(\frac{3}{2}\right)^{\lceil \log n \rceil}\right)$$

$$= \mathbf{9n} \times \left[\frac{\left(\frac{3}{2}\right)^{\lceil \log n \rceil + 1} - 1}{\frac{3}{2} - 1} \right] = 9n \times (2) \times \left[\left(\frac{3}{2}\right)^{\lceil \log n \rceil + 1} - 1 \right]$$

$$= 2 \times 9n\left[2^{\log_2 \frac{3}{2}}\right]^{\log n+1} - 18n = 18n\left[2^{(\log_2 3\ -1)}\right]^{(\log n+1)} - 18$$

$$= 18n\left[2^{((\log_2 n)^\wedge \log_2 3\ - \log_2 n + \log_2 3\ -1)}\right] - 18n = 18n\left[\frac{n^{\log_2 3} \times 2^{\log_2 3\ -1}}{n}\right] - 18n$$

$$= 18 \times 2^{(\log_2 3\ -1)} \times n^{\log_2 3} - 18n = o(n^{\log_2 3})$$