L2

aug 29 2013

shelat

warmup

Simplify $(1 + a + a^2 + \cdots + a^L)(a - 1) =$ $a^{L+1} - 1$

$a + a^2 + a^3 + \cdots + a^L + a^{L+1}$

$-1 - a - a^2 - a^3 - \cdots - a^L$

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

warmup

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

hw0 submission

https://church.cs.virginia.edu/13f4102

1. stand
2. set your “number” to one
3. greet a neighbor (pause if odd person out)
4. if you are older, give your “number” to young and sit
   if you are younger, add “numbers”
5. if you are standing & you have a neighbor, goto 3

1 stand
2 set
3 greet
4 sit/add
5 repeat

how fast does it work:

$T(n)$: running time on an instance w/ n people.

1. stand
2. set
3. greet
4. sit/add
5. repeat

how fast does it work:

$$T(n)$$

time to finish for a room of size n

1 stand 2 set 3 greet 4 sit/add 5 repeat

how fast does it work:

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

$T(1) = 3$

recurrence?

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

$$T(1) = 3$$

solve a simpler case when n is a power of 2.

$$T(2^k) = 2 + T(2^{k-1})$$

$$2 + T(2^{k-2})$$

$$2 + T(2^{k-3})$$

$$= 2 + \underbrace{(2 + 2 + \dots + 2)}_{k-1} + \underbrace{T(2^0)}_{T(1) = 3}$$

$$= 2k + 3 = 2 \cdot \log_2(2^k) + 3$$

$$T(n) \le T\left(2^{\lceil \log_2 n \rceil}\right) = 2 \cdot \lceil \log_2 n \rceil + 3$$

$$T(2^k) = 2 + T(2^{k-1})$$

$$= 2 + 2 + T(2^{k-2})$$

"intuition here"

$$= 2 + \overbrace{2 + \cdots + 2}^{k-1} + T(2^0)$$

$$= 2k + 3$$

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

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

Asymptotic notation

$O(f)$ at most within const of f for large n

(set of functions

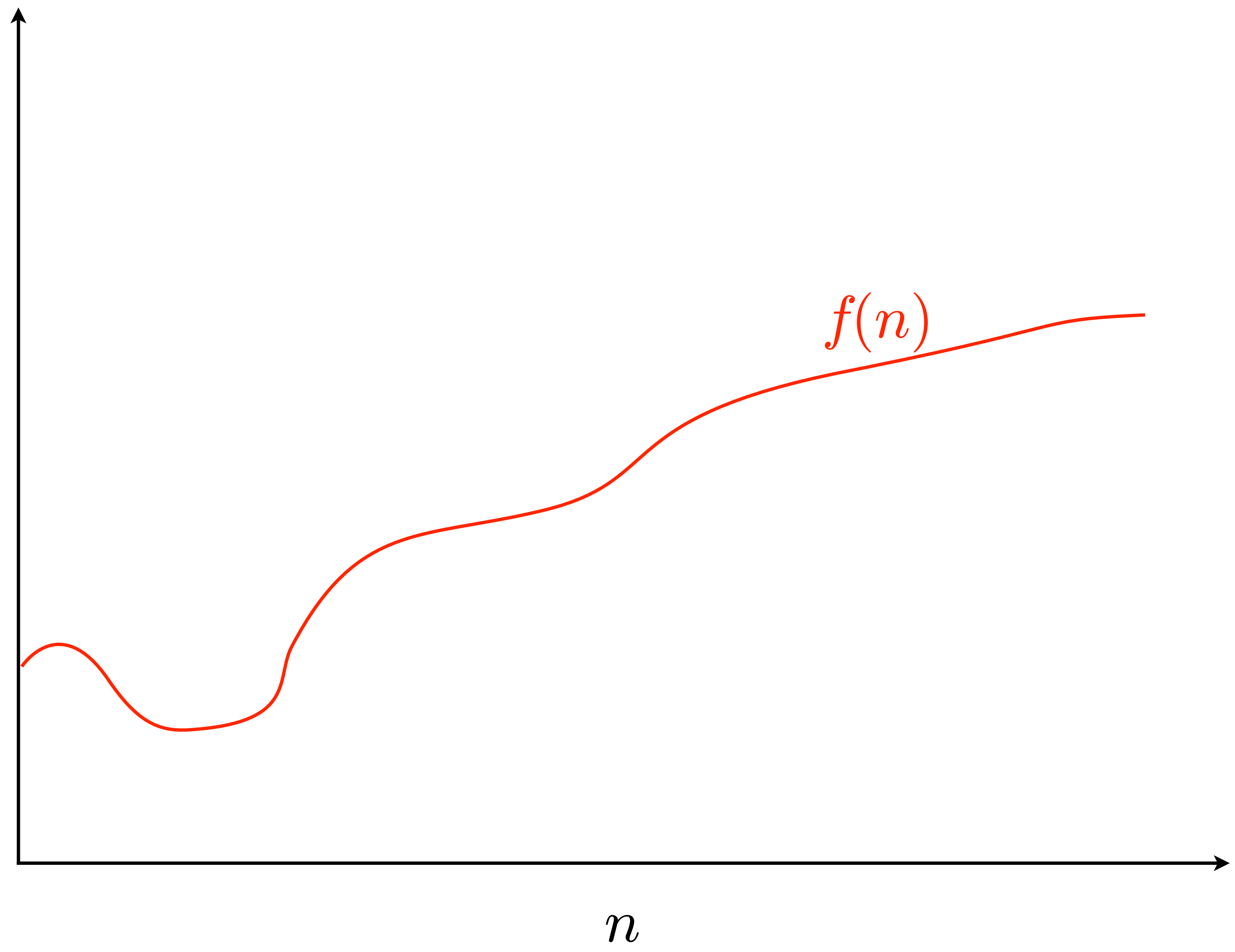
$O(f) = \{ g \mid \exists n_0, c$ such that for all $n > n_0$ $g(n) < c \cdot f(n) \}$

Asymptotic notation

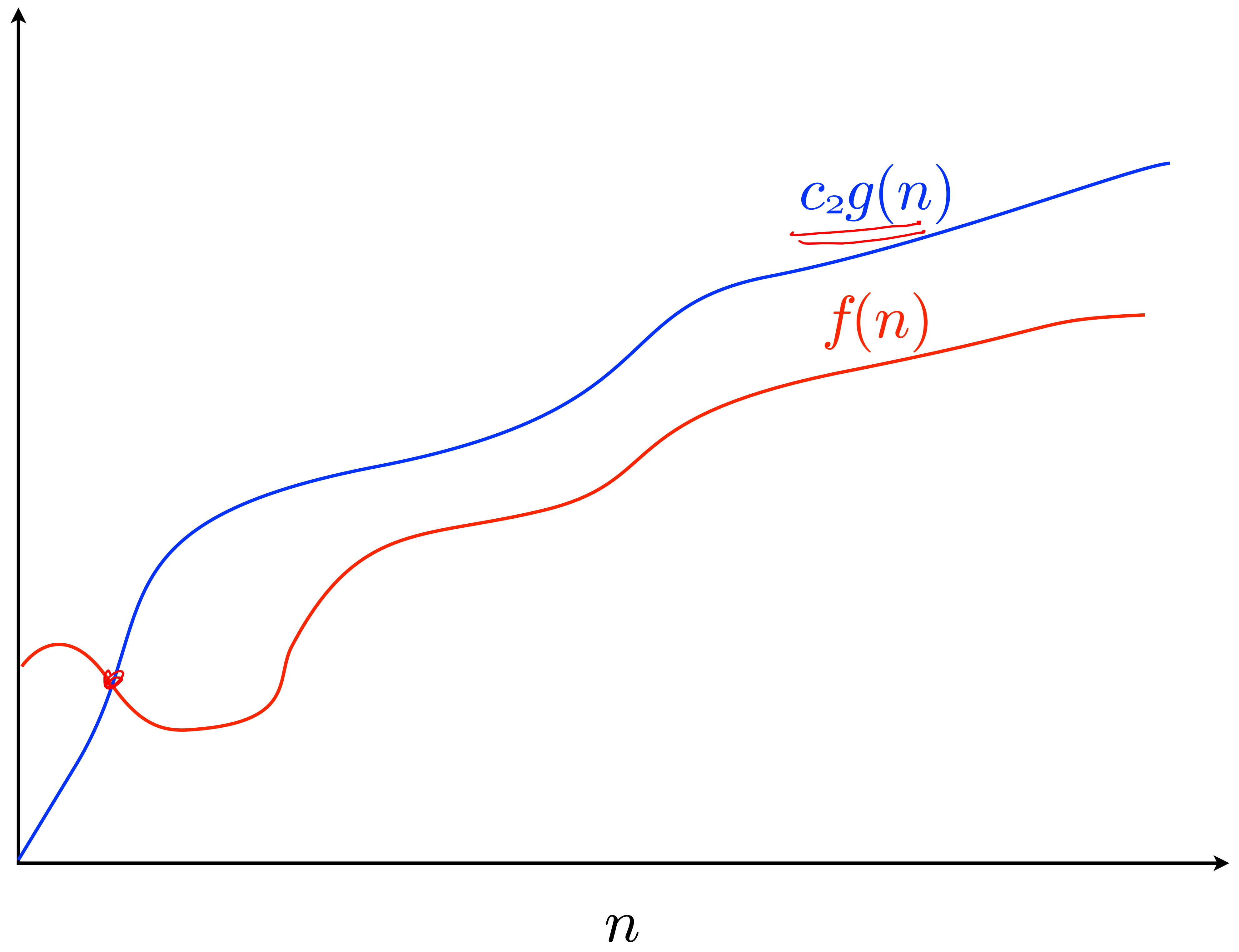
$O(f)$ at most within const of f for large n

asymptotic notation

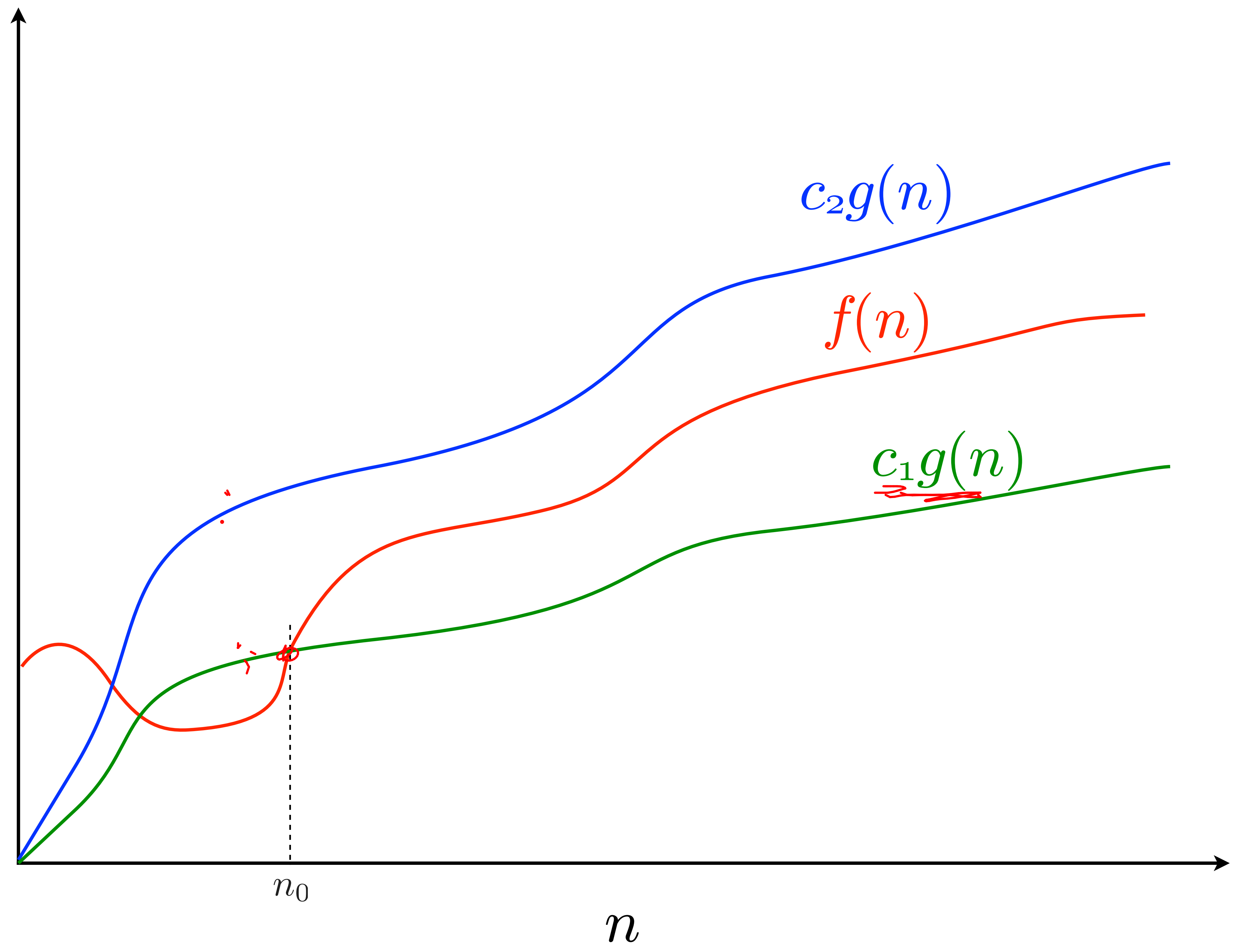
$O(f)$	at most within const of f	for large n
$\Omega(f)$	at least within const of f	for large n
$\Theta(f)$	within a const of f	for large n



$f(n)$
n



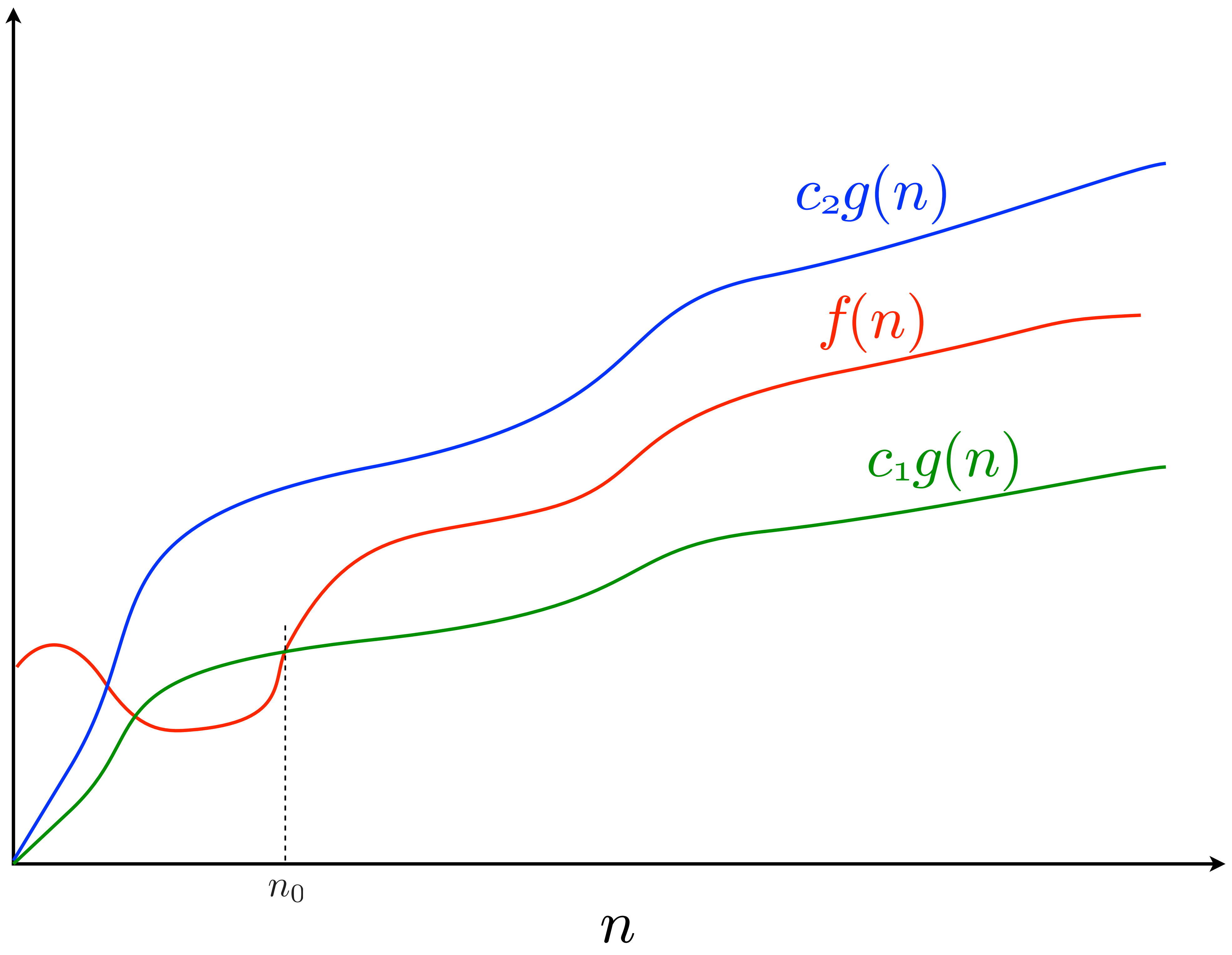
$c_2g(n)$
$f(n)$
n



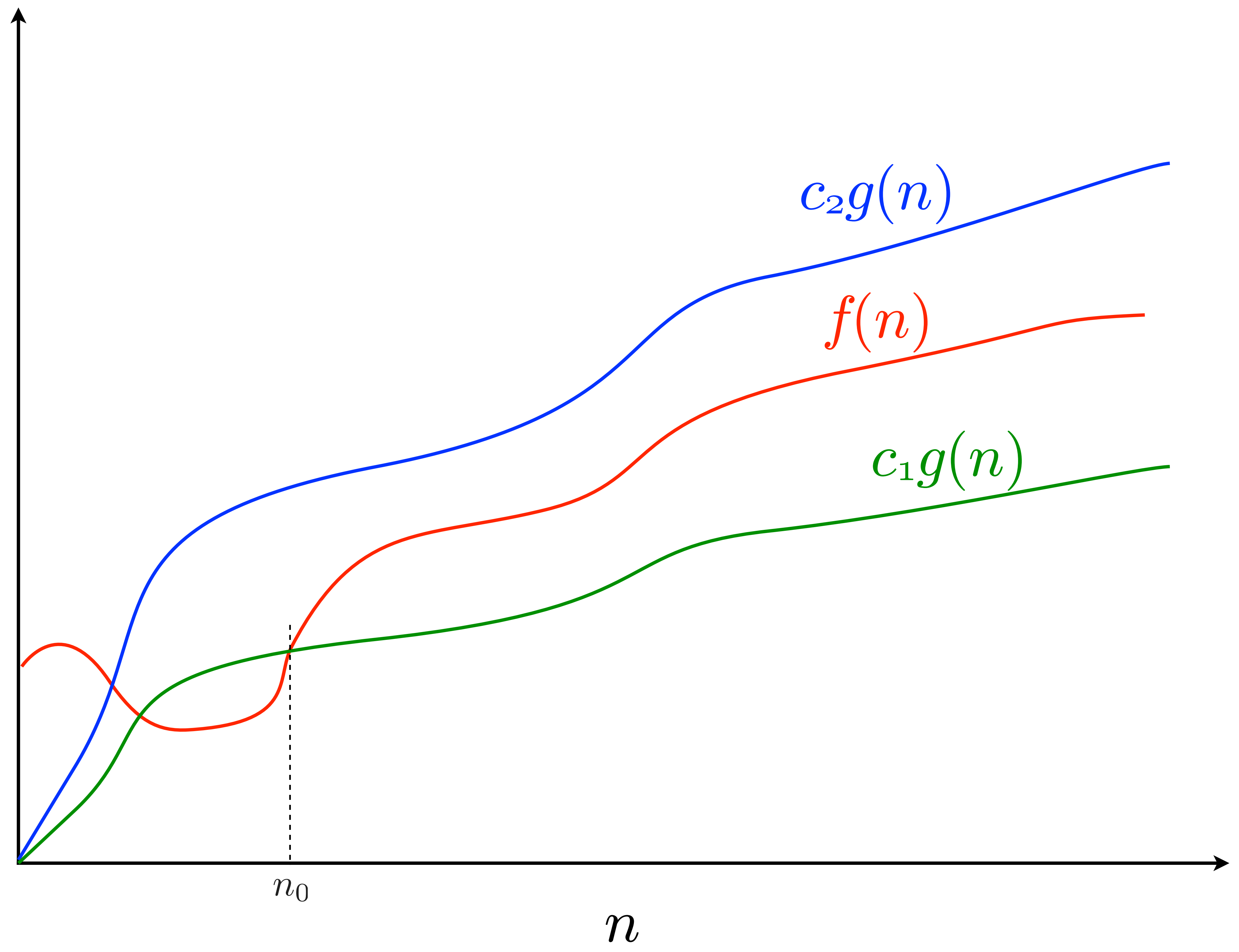
$f \in \Theta(g)$

$f = \Theta(g)$

function ↑ ↑ set

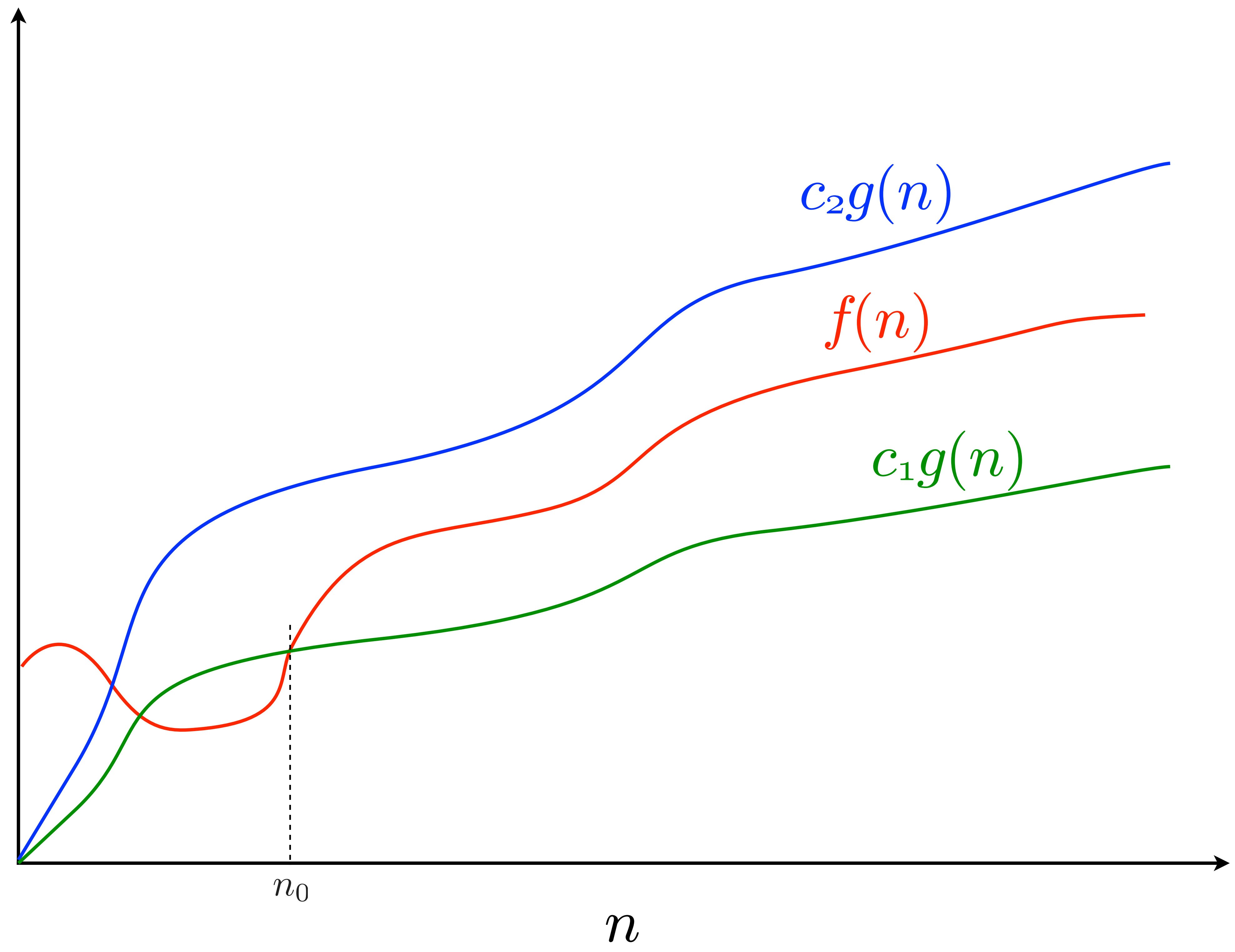


$$f(n) = O(g(n))$$



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

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



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

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

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

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

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

?

actually, we have only shown that

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

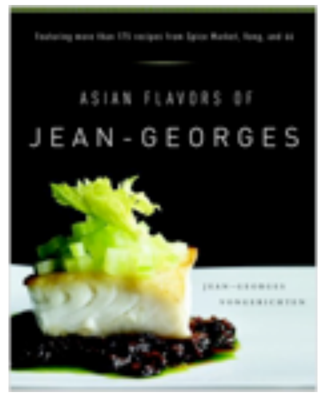
main ideas:

1. Theme 1: Solve big problems by making them into smaller ones.
2. Analyze performance of the algorithm using recurrence
3. We removed unimportant details thru the use of Asymptotic notation

How to solve recurrence relations

 – tree

?-✓ guess & check (INDUCTION)



– cookbook (Master's thm)

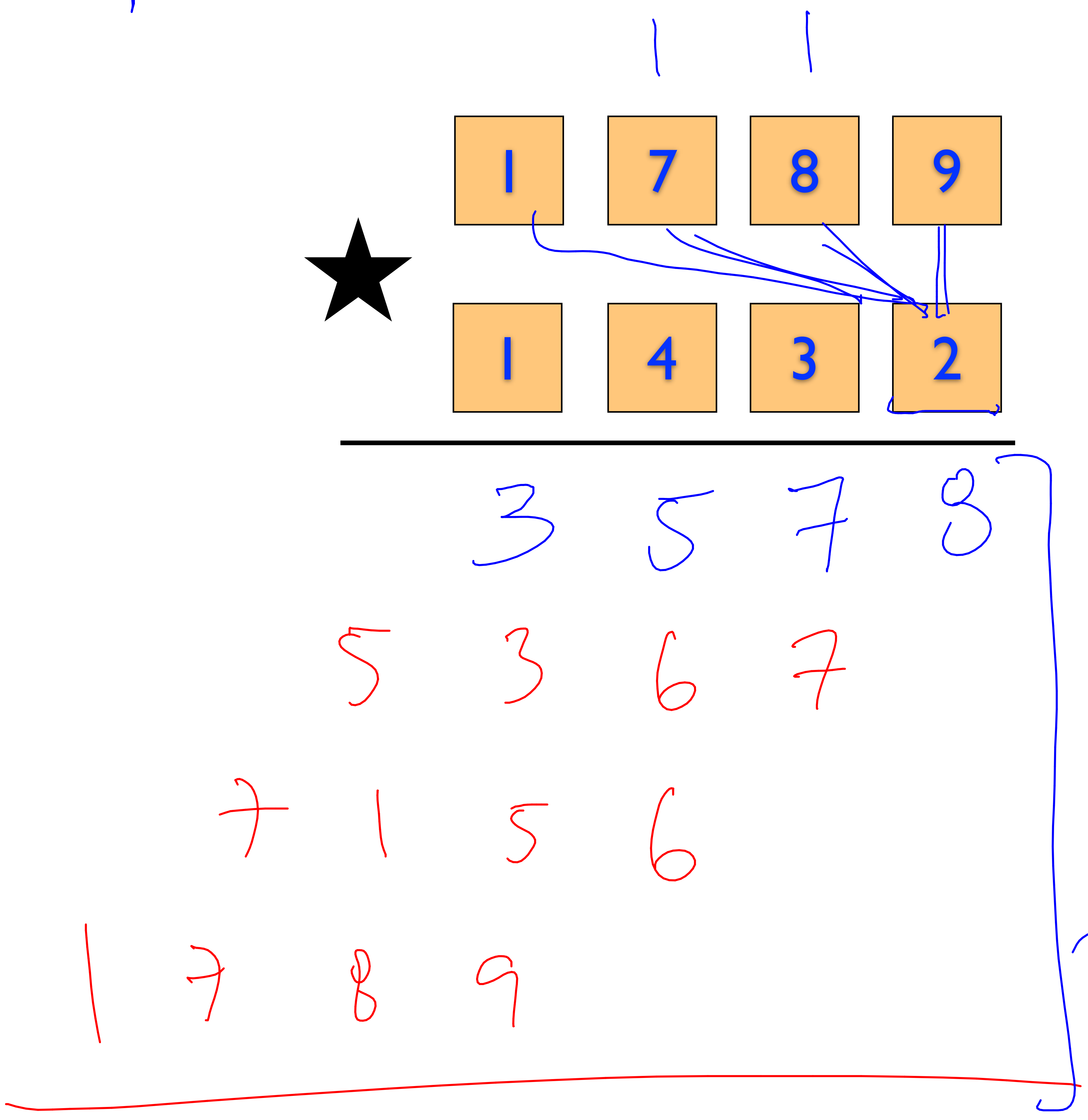
– substitution

http://www.drblank.com/law301.jpg

Multiplication

Input: 2 n-digit numbers

Basic operation is $d \times d$



Total work: $n(2n-1) = \Theta(n^2)$

$(n\text{-}1)(n+1)$ +

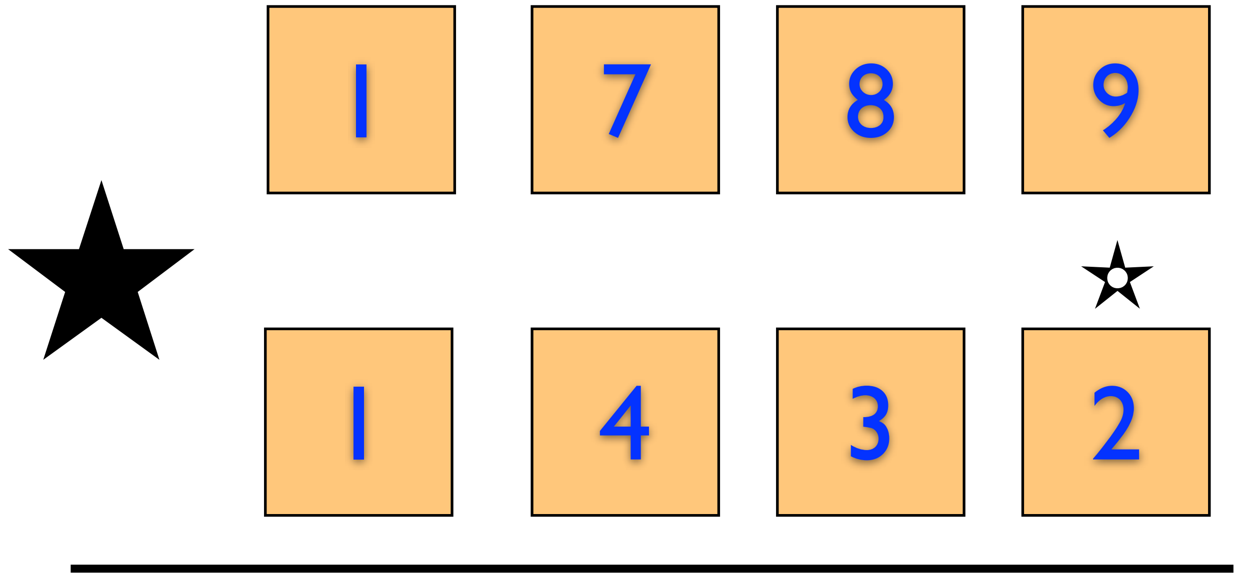
→ n multiplications, n−1 adds

→ n mults, n−1 adds

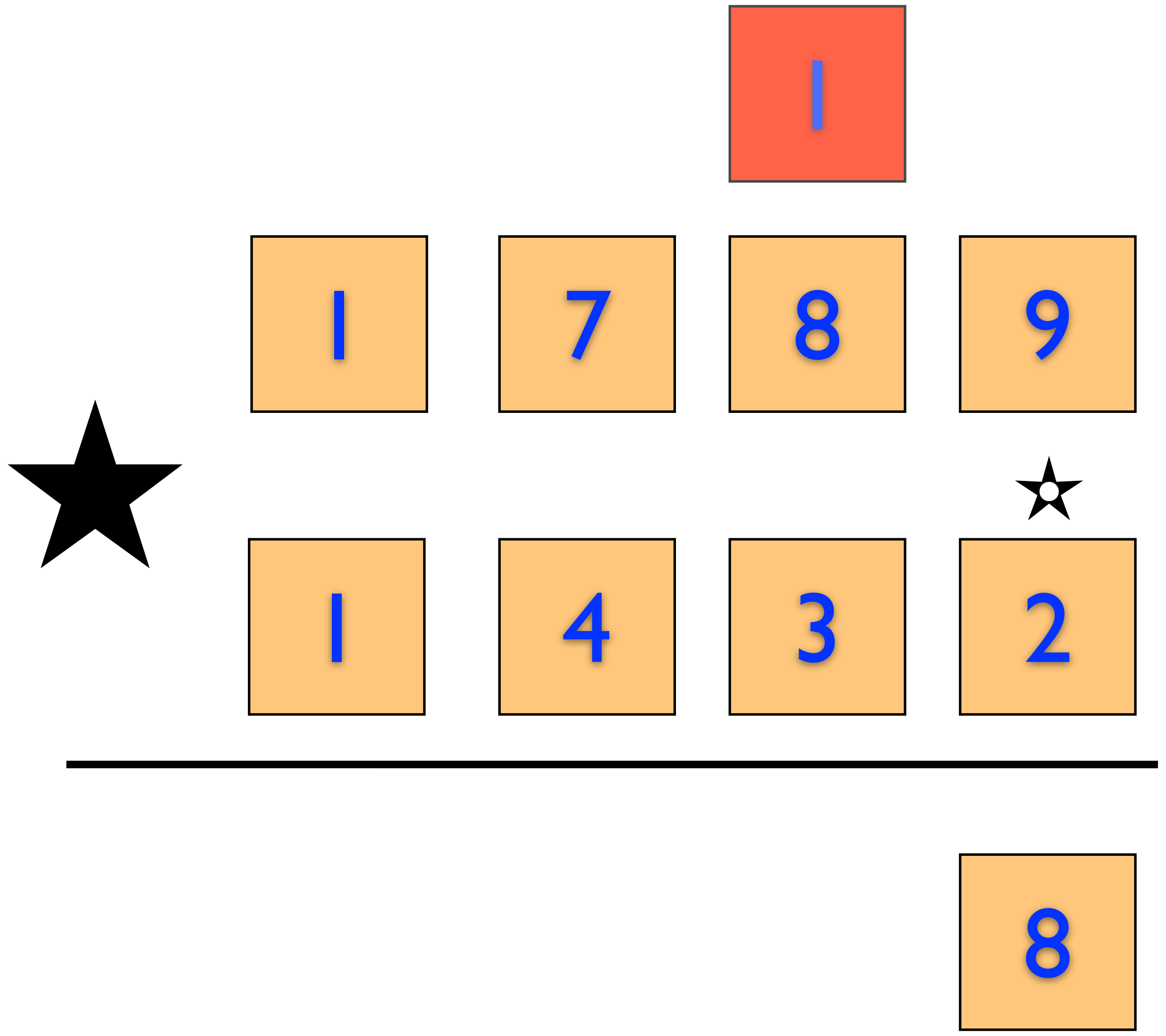
→ n mults ''

→ '' ''

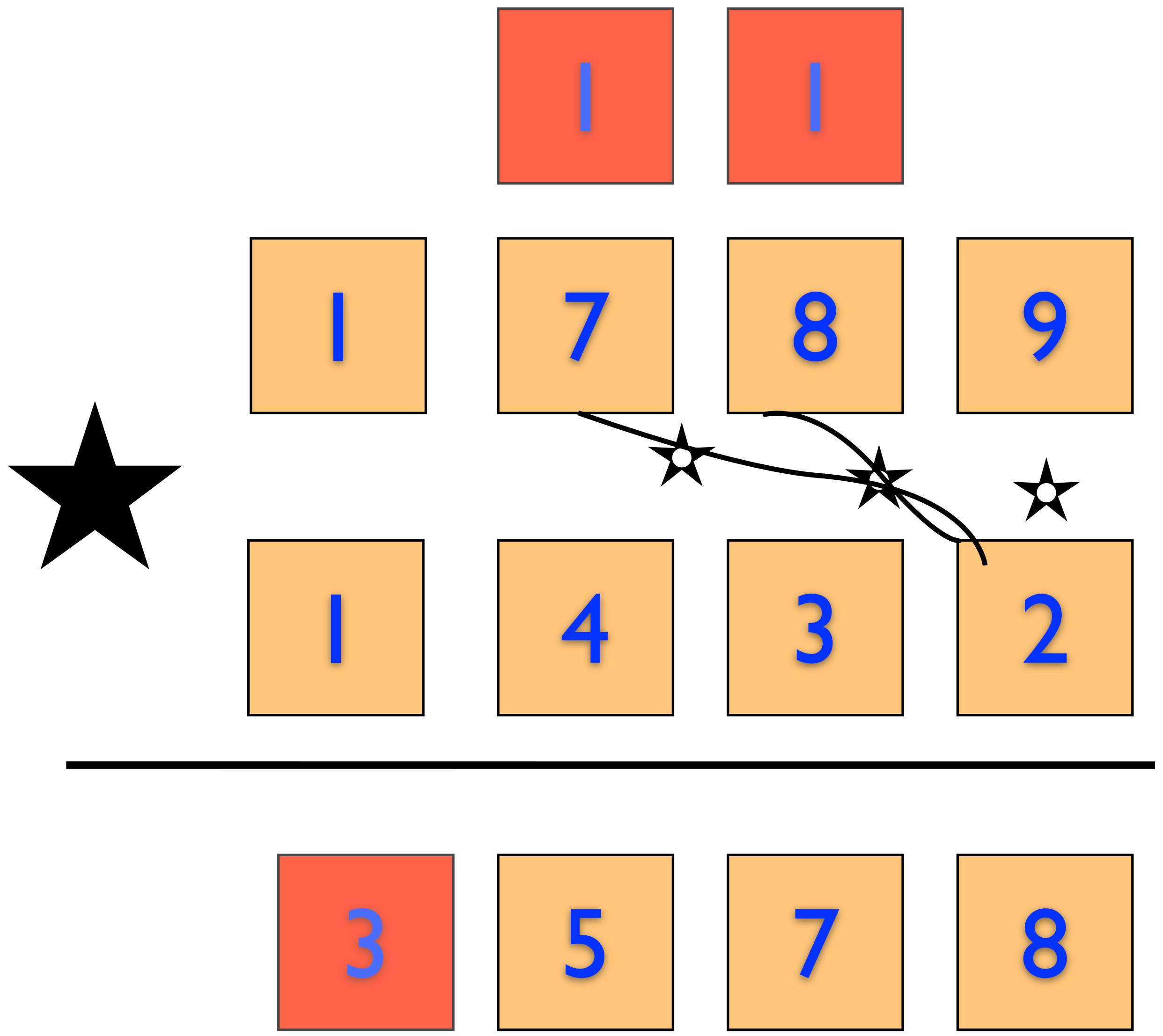
n lines



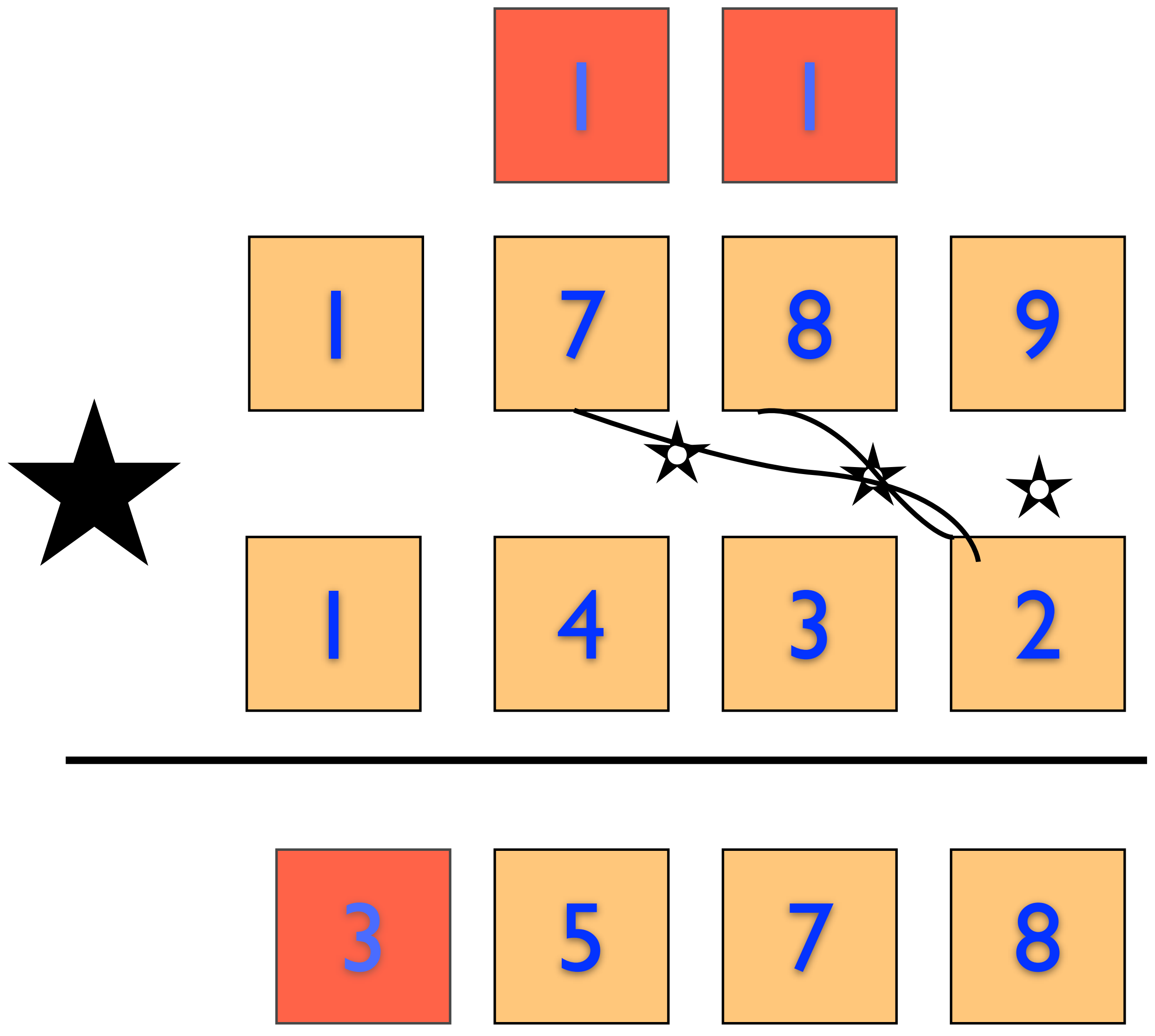
$(n\text{-}1)(n+1)$ ✛



$(n\text{-}1)(n+1)$ +

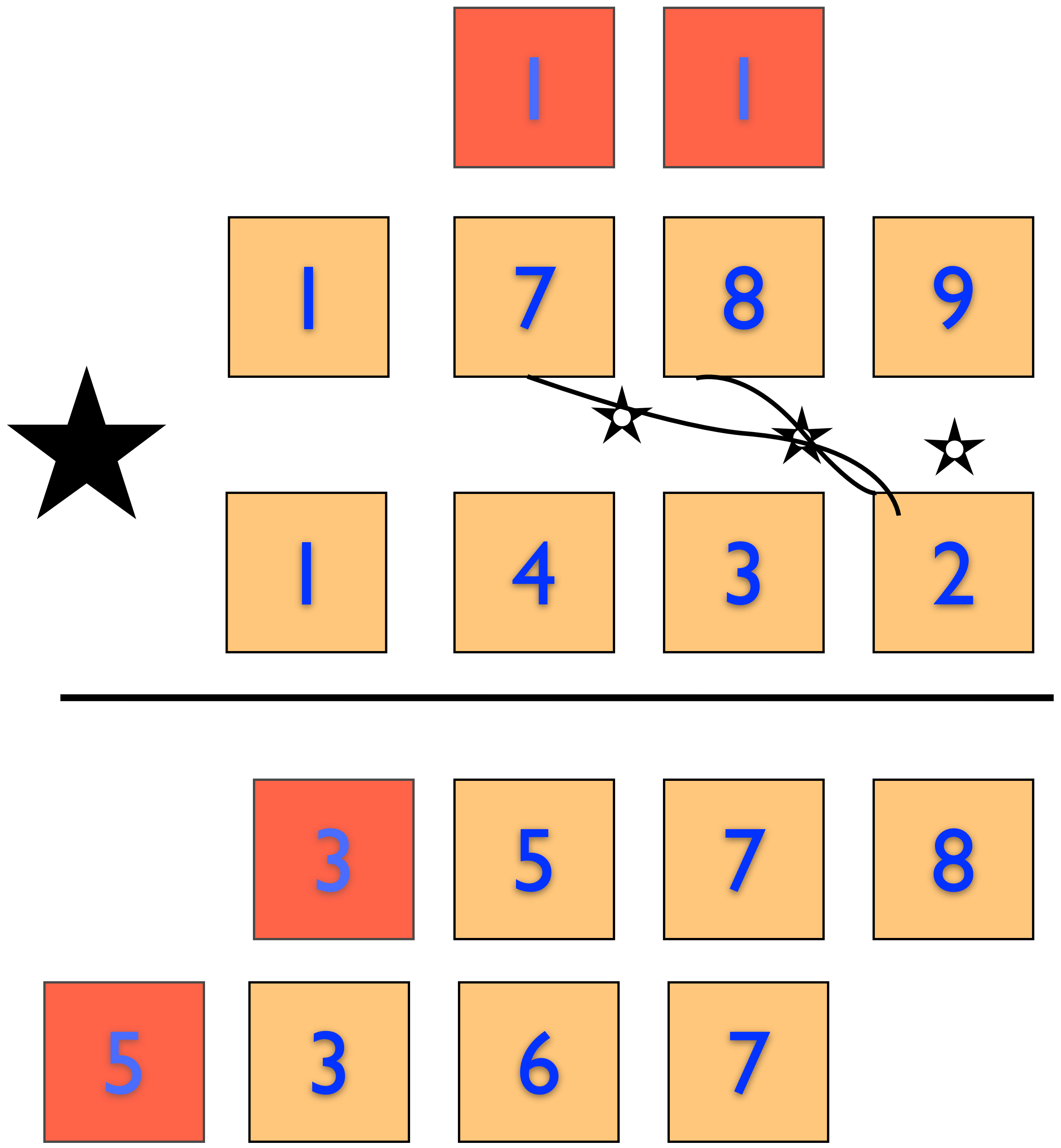


$(n\text{-}1)(n+1)$ +



$(n-1)(n+1)$ ✣

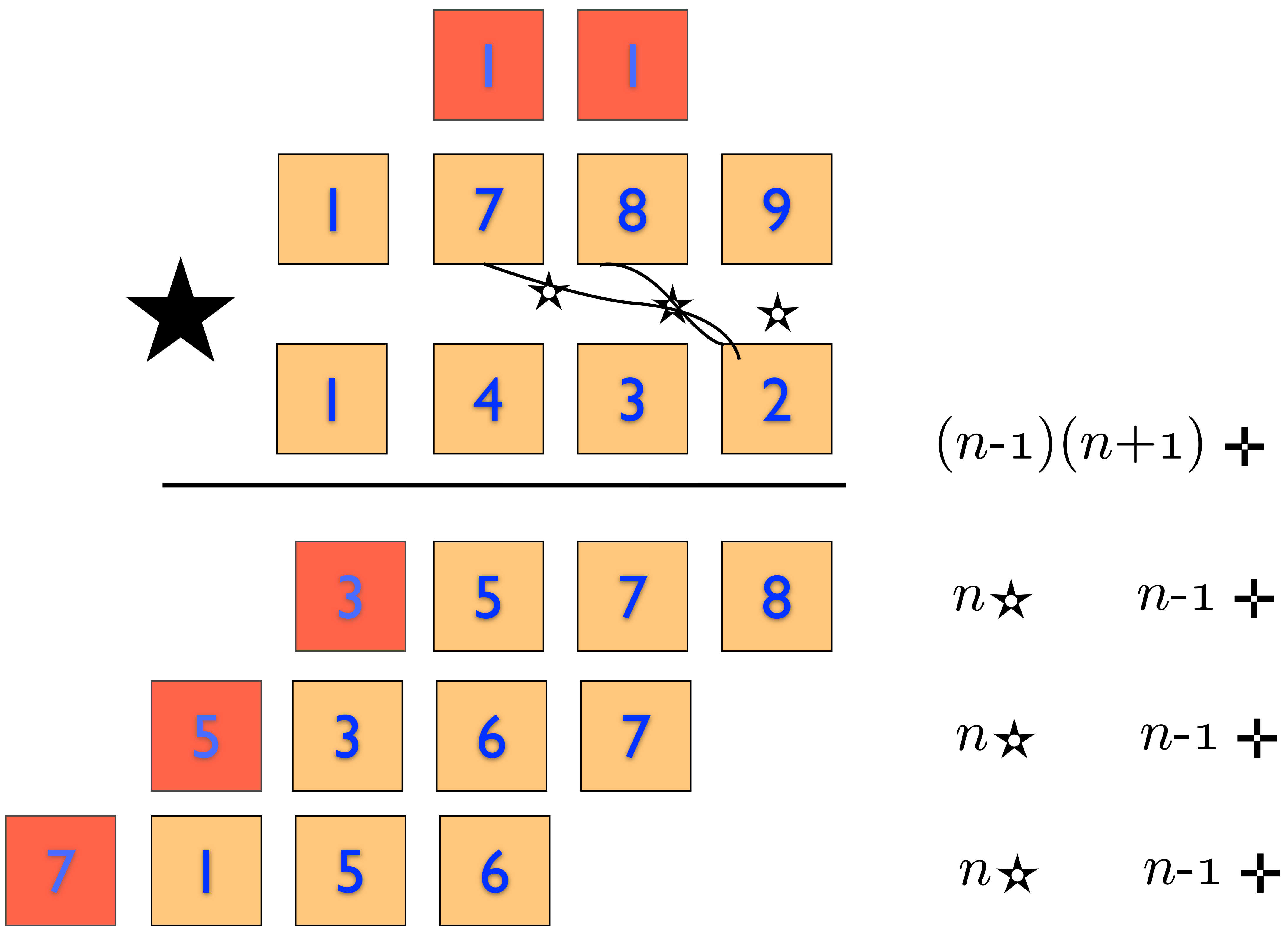
n ✫ $n-1$ ✣



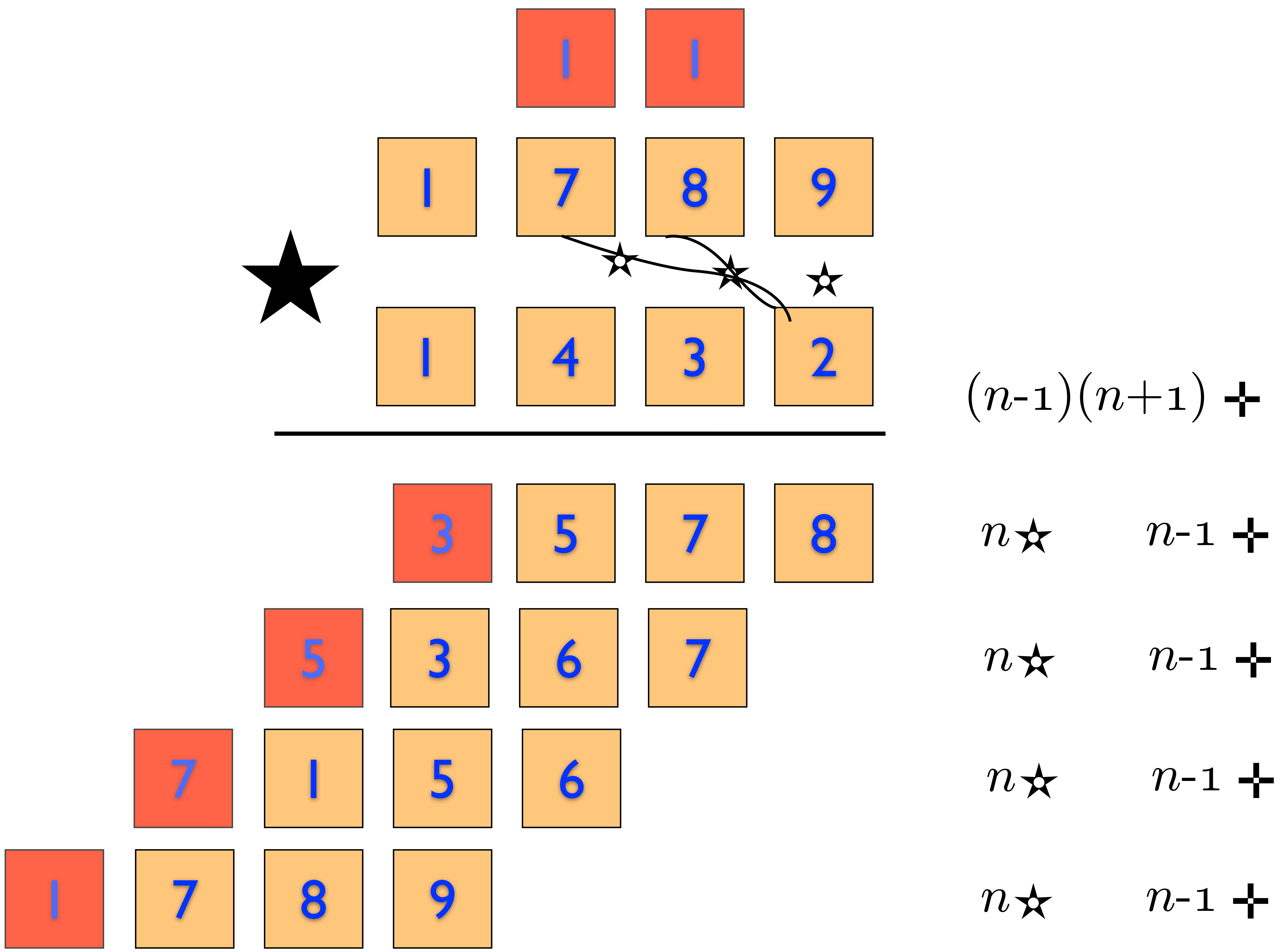
$(n-1)(n+1)$ ✢

n ✫ $n-1$ ✢

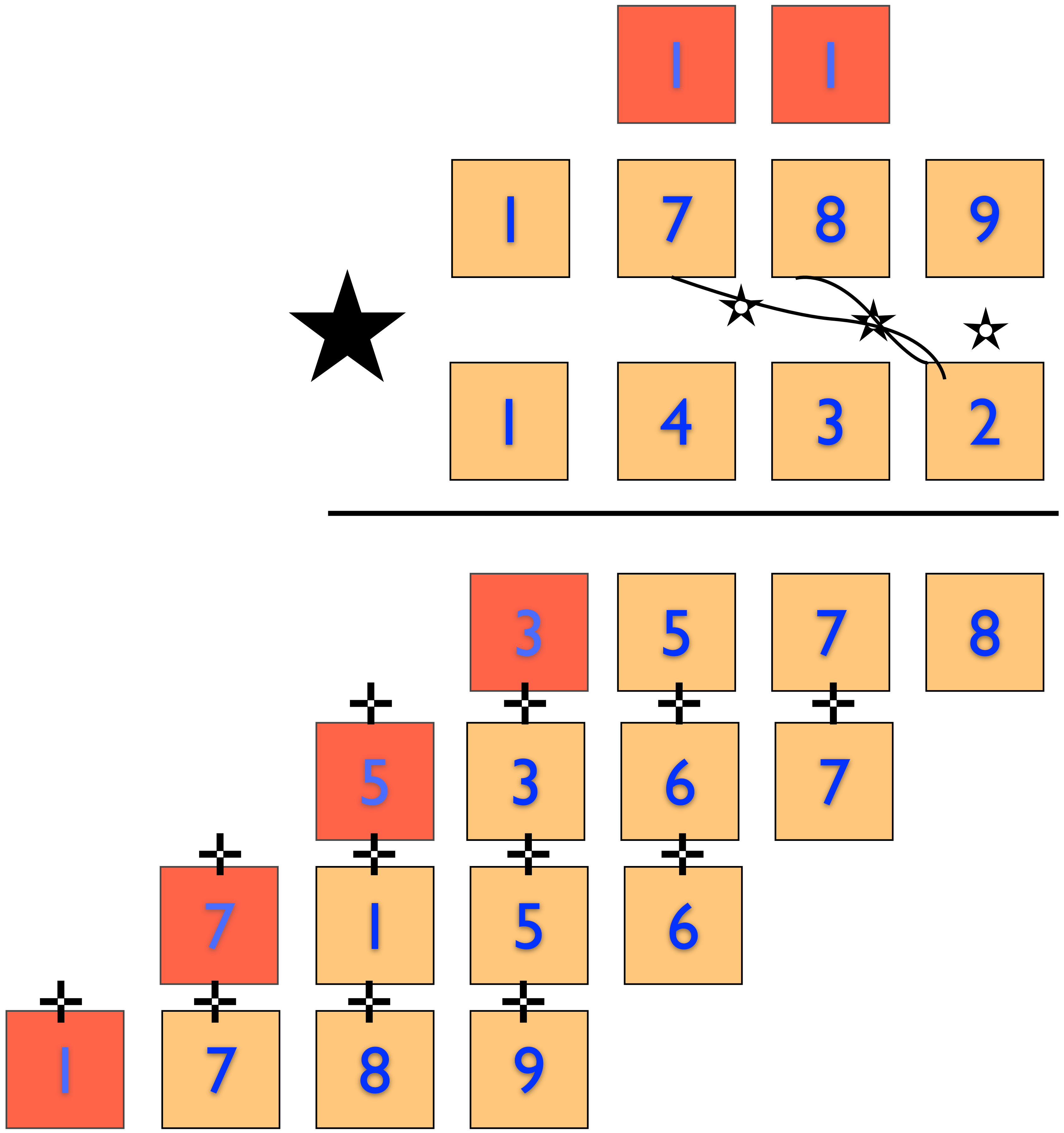
n ✫ $n-1$ ✢



1 1
1 7 8 9
1 4 3 2
(n-1)(n+1)
3 5 7 8
n n-1
5 3 6 7
n n-1
7 1 5 6
n n-1



1 1
1 7 8 9
1 4 3 2
(n-1)(n+1)
3 5 7 8
n n-1
5 3 6 7
n n-1
7 1 5 6
n n-1
1 7 8 9
n n-1



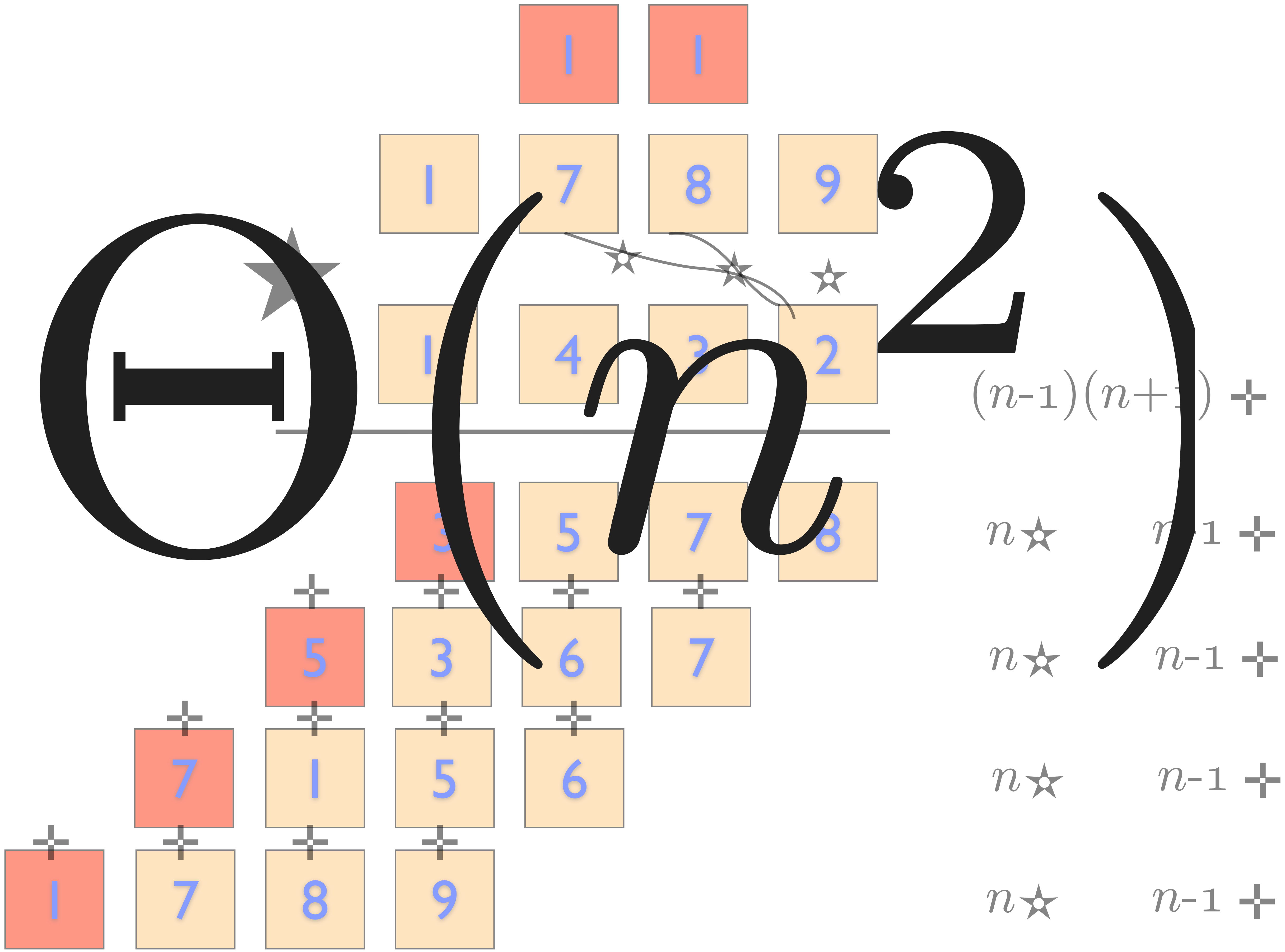
$(n-1)(n+1)$ ✣

n ☆ $n-1$ ✣

n ☆ $n-1$ ✣

n ☆ $n-1$ ✣

n ☆ $n-1$ ✣


Θ(n²)
(n-1)(n+1)
n
n-1
n
n-1
n
n-1
n
n-1

Theme 1

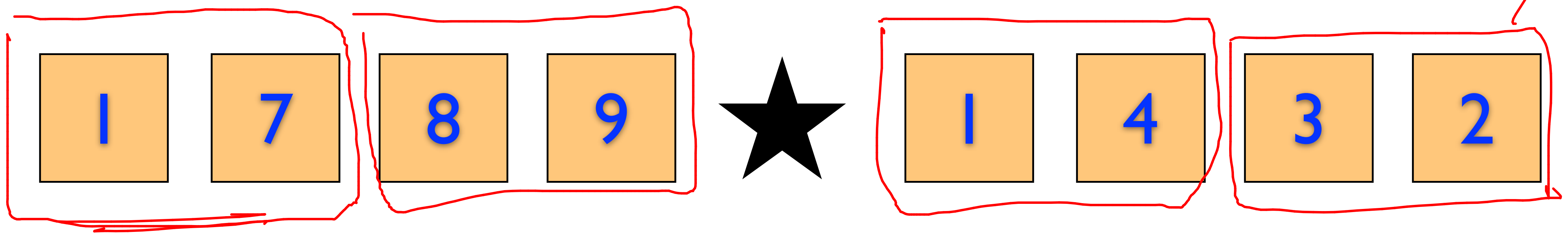
DIVIDE & conquere for mult

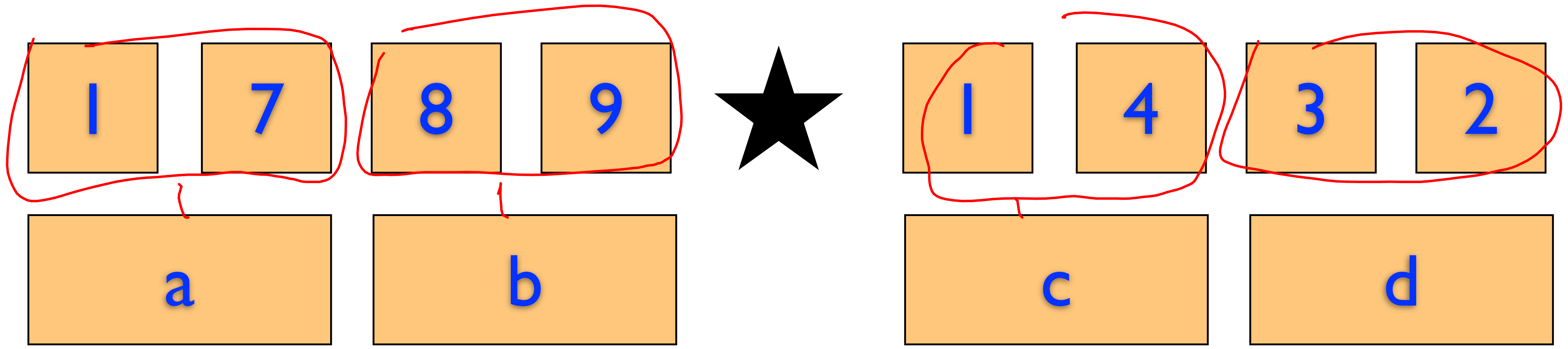
FORMULATE THE MULT OF N-DIGIT

NUMBERS INTO SMALLER INSTANCE.

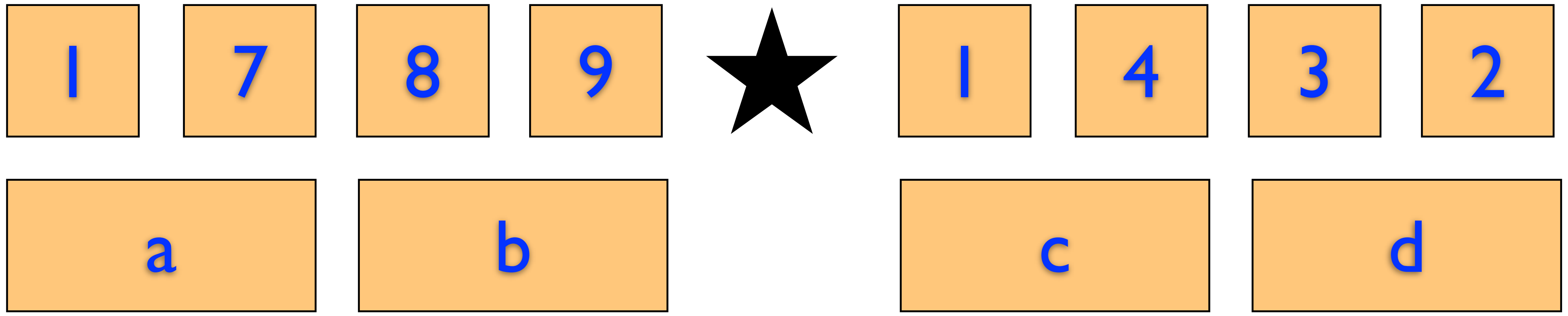
$(17 \cdot 100 + 89) \cdot (14 \cdot 100 + 32)$

→ n/2 digit numbers.





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



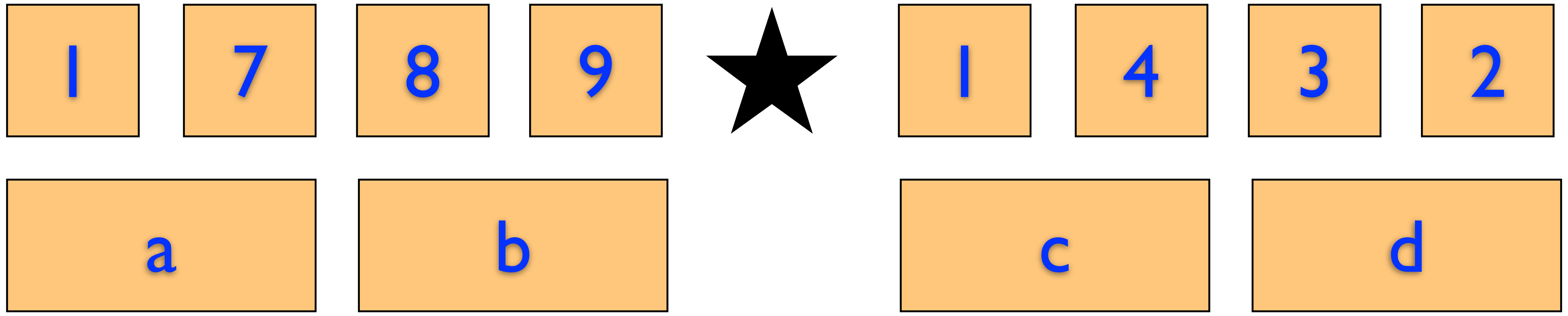
$$ac100^2 + (ad + bc)100 + bd$$

1 7 8 9 ★ 1 4 3 2

a b c d

$$ac100^2 + (ad + bc)100 + bd$$

4✫ 3✛



$$ac100^2 + (ad + bc)100 + bd$$

4★ 3+

$$T(n) = 4T(n/2) + 3O(n)$$

Mult(ab, cd) = T(n)

→ base case of 1 digit.

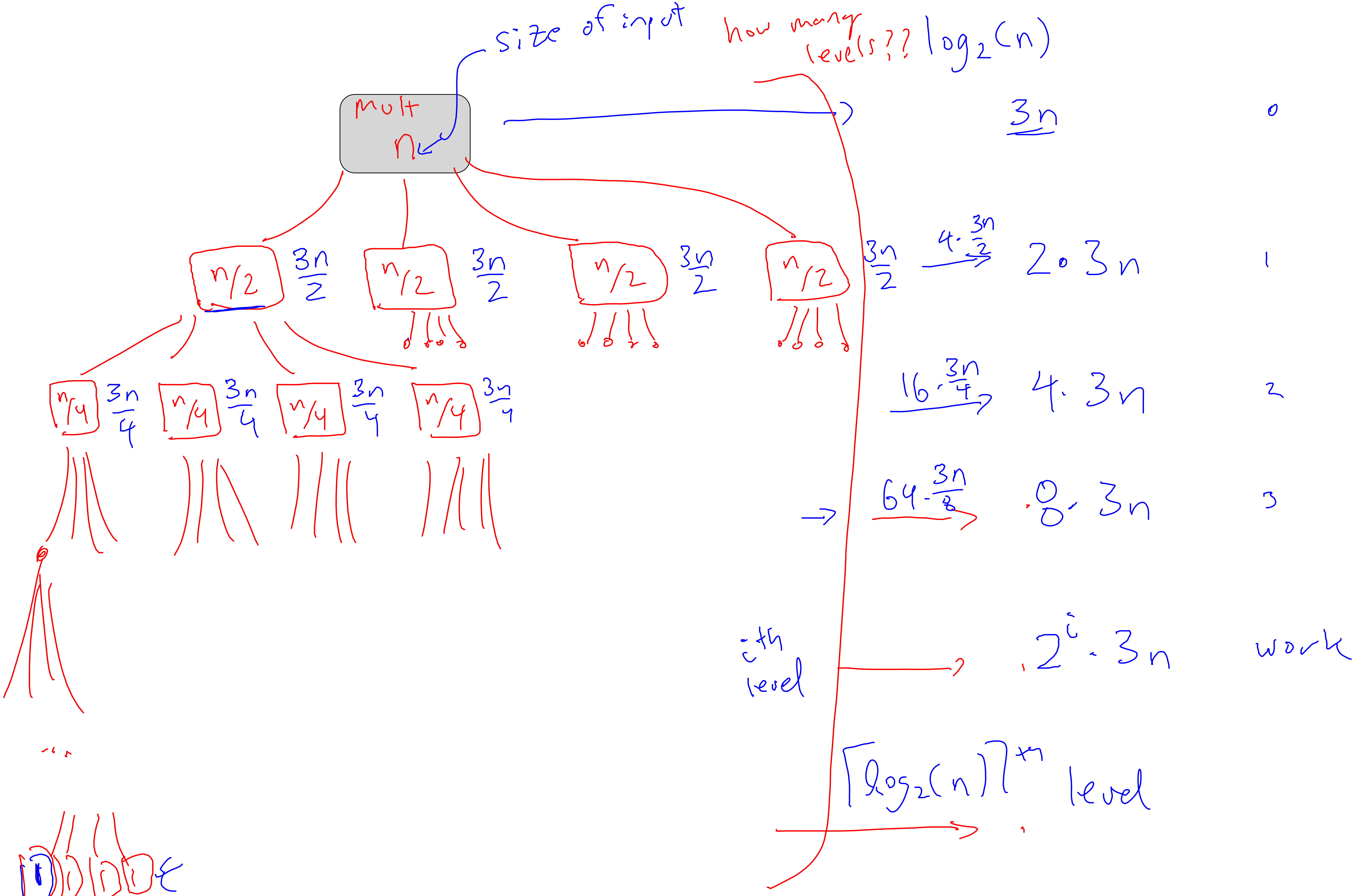
- Compute x = Mult(a,c) mult of 2 n/2 digit numbers → T(n/2)
- Compute y = Mult(a,d) → T(n/2)
- Compute z = Mult(b,c) →
- Compute w = Mult(b,d) →

Return r = x*100^2 + (y+z)100 + w

3 ADDITIONS O(n)

adding zeroes to the end

$$T(n) = 4T\left(\frac{n}{2}\right) + 3n$$


size of input
how many levels?? $\log_2(n)$
mult
n
n/2
$\frac{3n}{2}$
n/4
$\frac{3n}{4}$
3n
0
$4 \cdot \frac{3n}{2}$
$2 \cdot 3n$
1
$16 \cdot \frac{3n}{4}$
$4 \cdot 3n$
2
$64 \cdot \frac{3n}{8}$
$8 \cdot 3n$
3
i-th level
$2^i \cdot 3n$
work
$\lceil \log_2(n) \rceil^{th}$ level

$$T(n) = 4T(n/2) + 3O(n)$$

calculations:

$$T(n) = 3n + 2 \cdot 3n + 2^2 \cdot 3n + \dots + 2^{\lceil \log_2 n \rceil} \cdot 3n$$

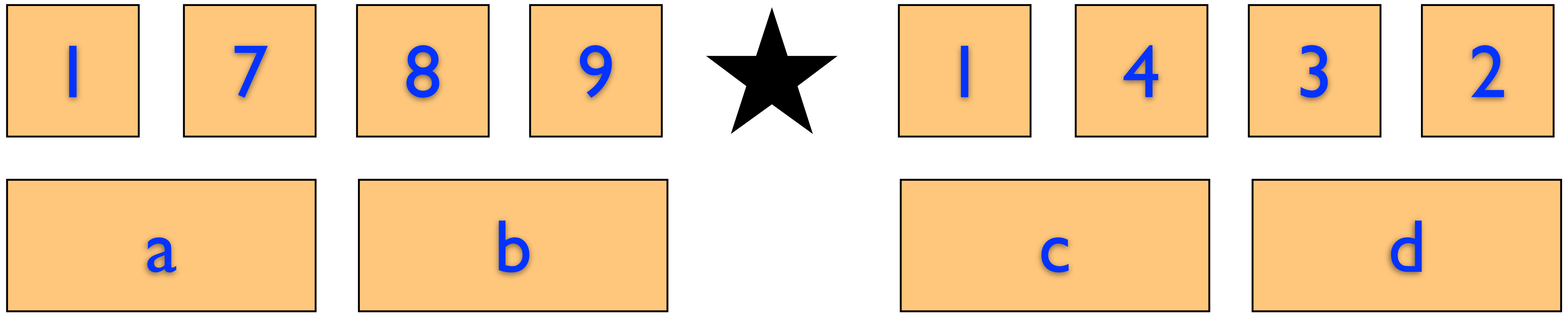
$$2n$$

$$= 3n\left(1 + 2 + 2^2 + \dots \frac{2^{\lceil \log_2 n \rceil}}{n}\right) n$$

$$\frac{a^{c+1} - 1}{a - 1} = \frac{2^{\lceil \log_2 n \rceil + 1} - 1}{1}$$

$$= 3n(2n - 1) = \Theta(n^2)$$

Karatsuba



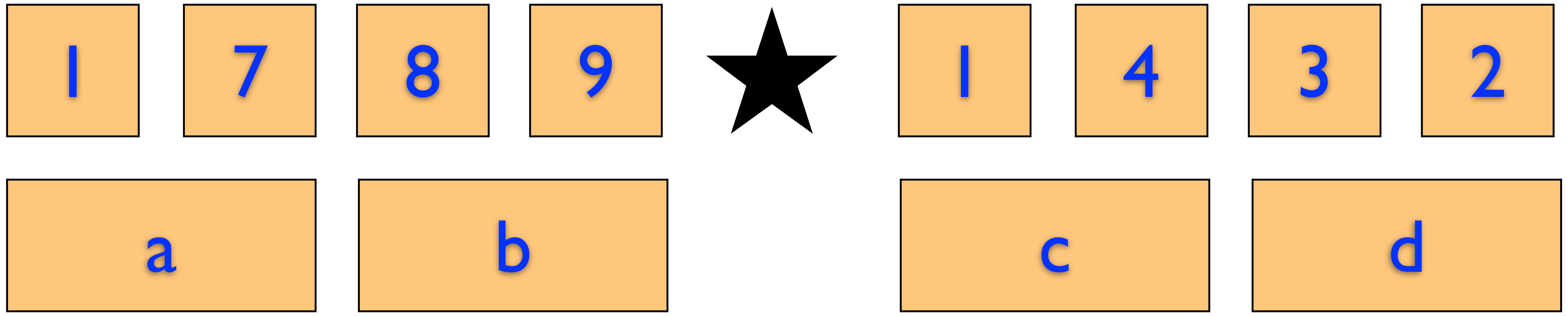
$$ac100^2 + (ad + bc)100 + bd$$

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

gold

subtract ac + bd from this term to leave (ad + bc)

Karatsuba

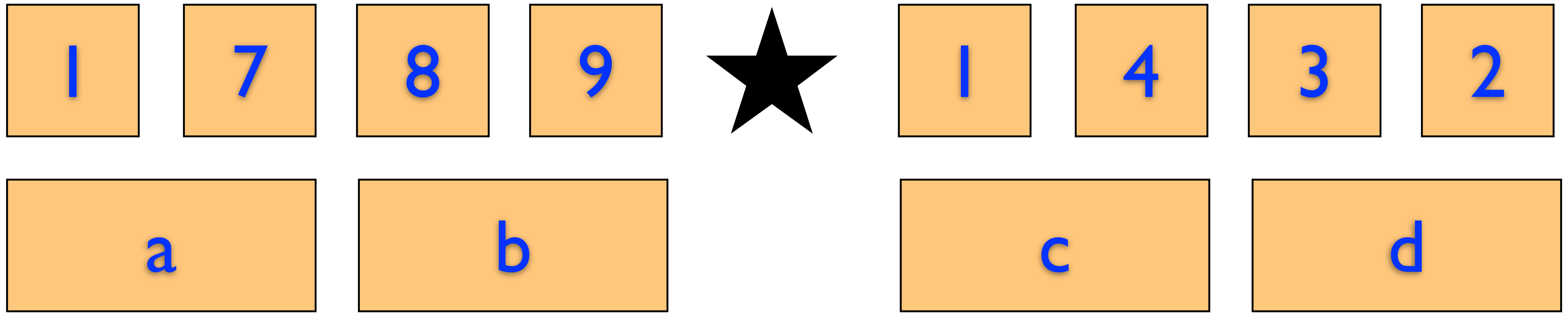


$$ac100^2 + (ad + bc)100 + bd$$

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

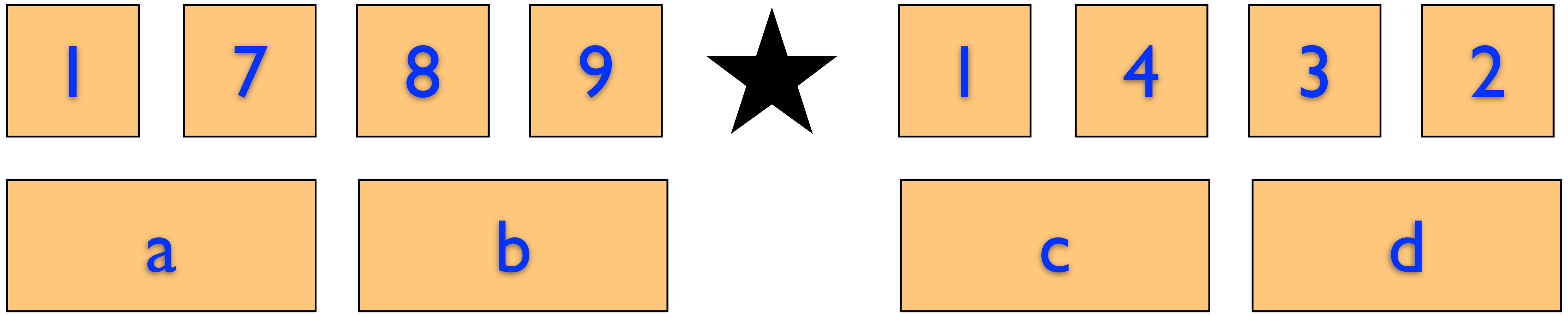
$$ad + bc = (a + b)(c + d) - ac - bd$$

Karatsuba algorithm



(1) mult(a,c) mult(b,d) mult((a+b),(c+d))

Karatsuba algorithm

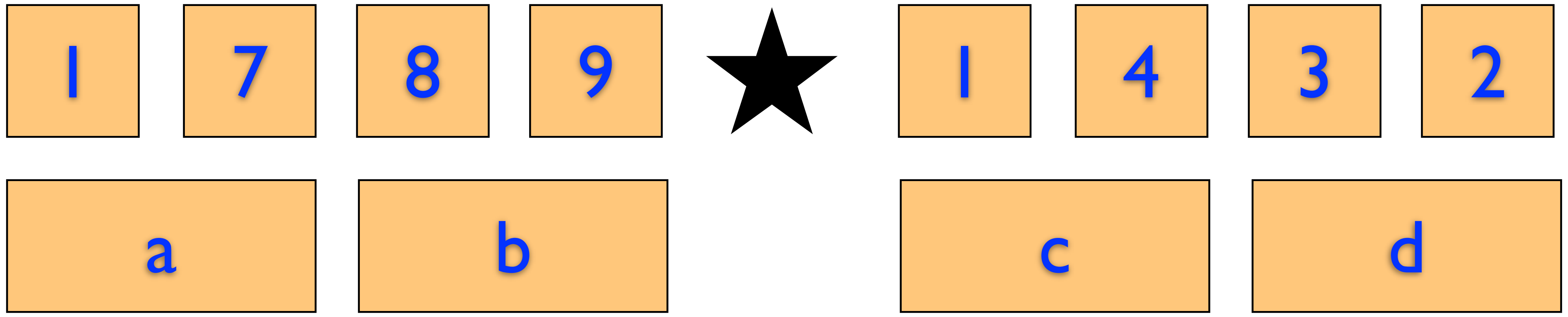


Recursively compute

1. $ac, bd, (a+b)(c+d)$

2. subtract off ac, bd from

Karatsuba algorithm

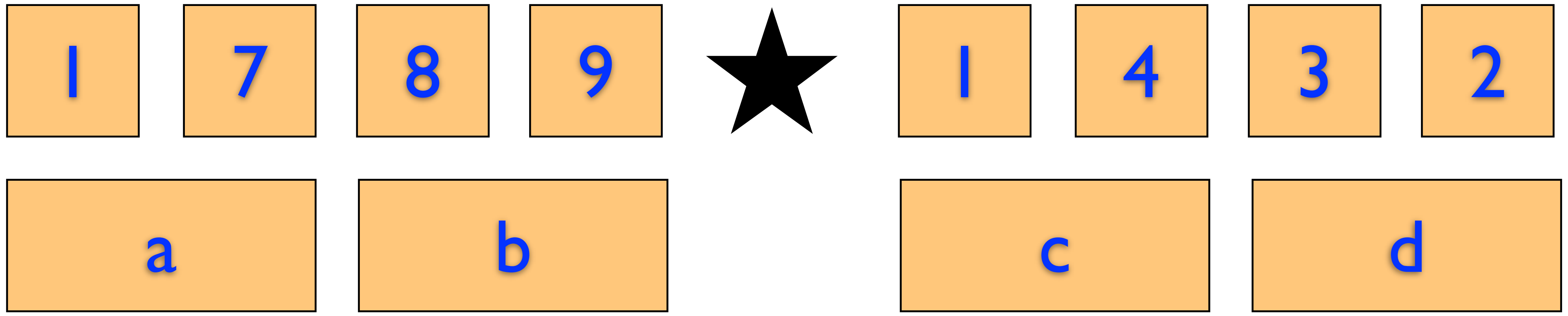


Recursively compute

❶ $ac, bd, (a+b)(c+d)$

❷ $ad + bc = (a+b)(c+d) - ac - bd$

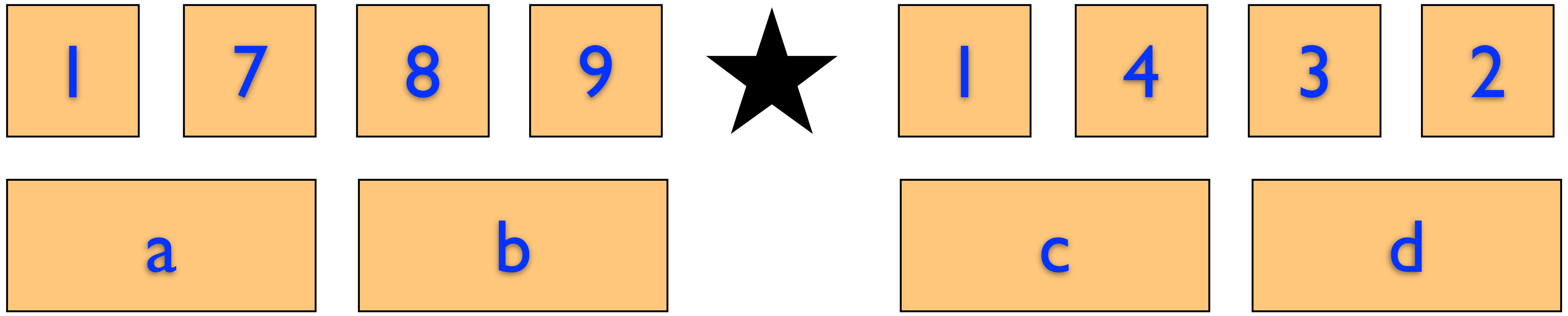
Karatsuba algorithm $T(n)$



Recursively compute

1. $ac, bd, (a+b)(c+d)$ → $3T(\frac{n}{2}) + 2n$
2. $ad + bc = (a+b)(c+d) - ac - bd$ → $+ \; 2n$
3. $ac100^2 + (ad+bc)100 + bd$ → $+ \; 2n$

Karatsuba algorithm



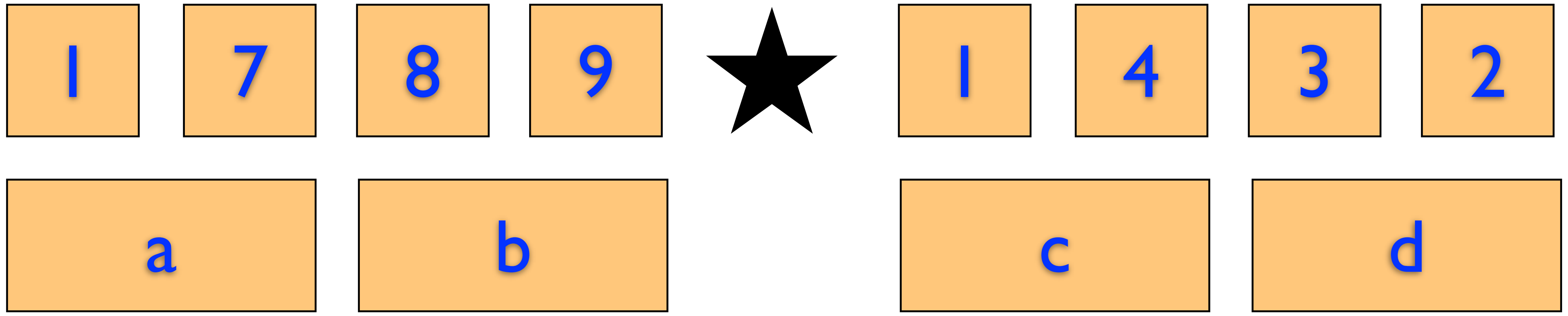
Recursively compute

❶ $ac, bd, (a+b)(c+d)$ $3T(n/2) + 2O(n)$

❷ $ad + bc = (a+b)(c+d) - ac - bd$ $2O(n)$

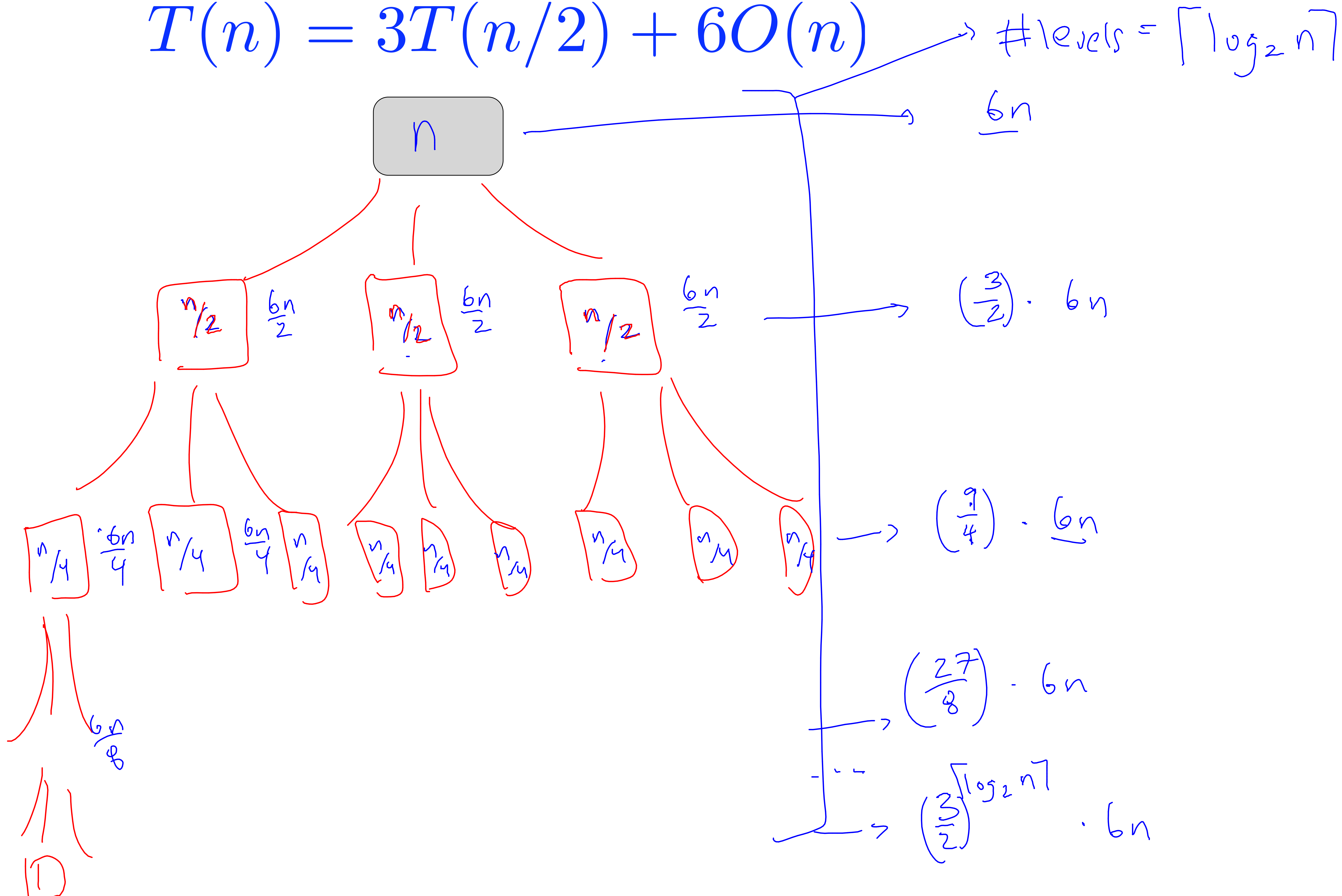
❸ $ac100^2 + (ad+bc)100 + bd$ $2O(n)$

Karatsuba algorithm



$$T(n) = 3T(n/2) + 6O(n)$$

$$T(n) = 3T(n/2) + 6O(n)$$


#levels = ⌈log₂ n⌉
n
6n
n/2
6n/2
n/4
6n/4
6n/8
1
(3/2) · 6n
(9/4) · 6n
(27/8) · 6n
(3/2)^⌈log₂ n⌉ · 6n

calculations:

$$6n\left(1+\left(\frac{3}{2}\right)+\left(\frac{3}{2}\right)^2+\cdots+\left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil}\right) = \frac{\left(\frac{3}{2}\right)^{\lceil \log_2 n \rceil + 1} - 1}{\frac{1}{2}}$$

$$\cdots$$

$$= O\left(n^{\log_2 3}\right)$$

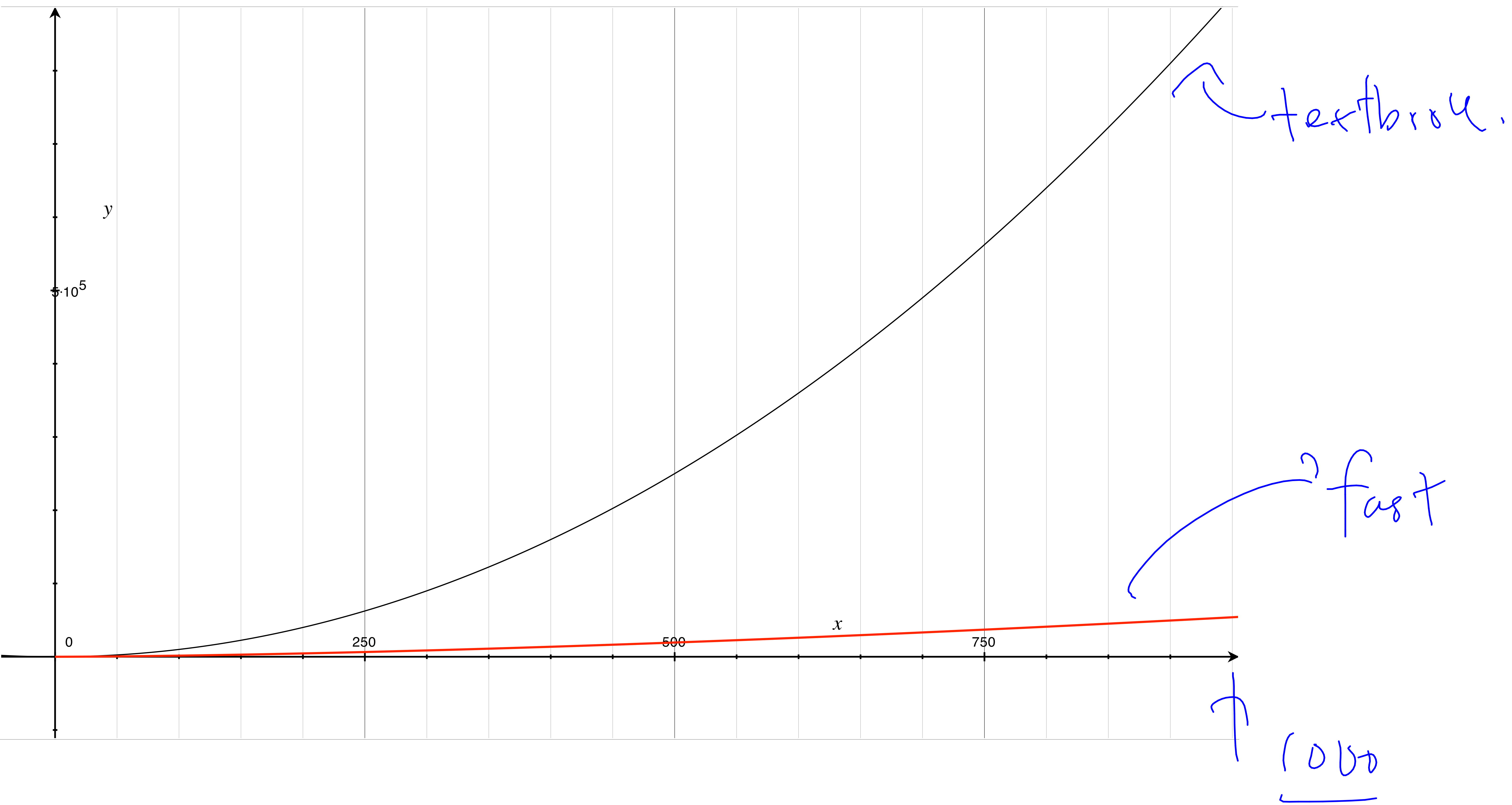
calculations:

$$T(n) = 3T(n/2) + 6O(n)$$

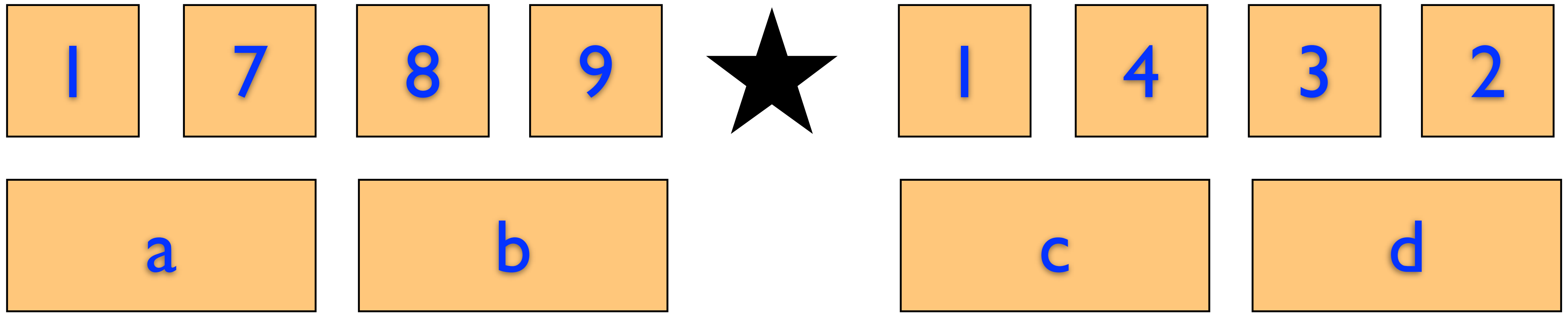
$$O(n^{\log_2(3)})$$

$$T(n) = 3T(n/2) + 6O(n)$$

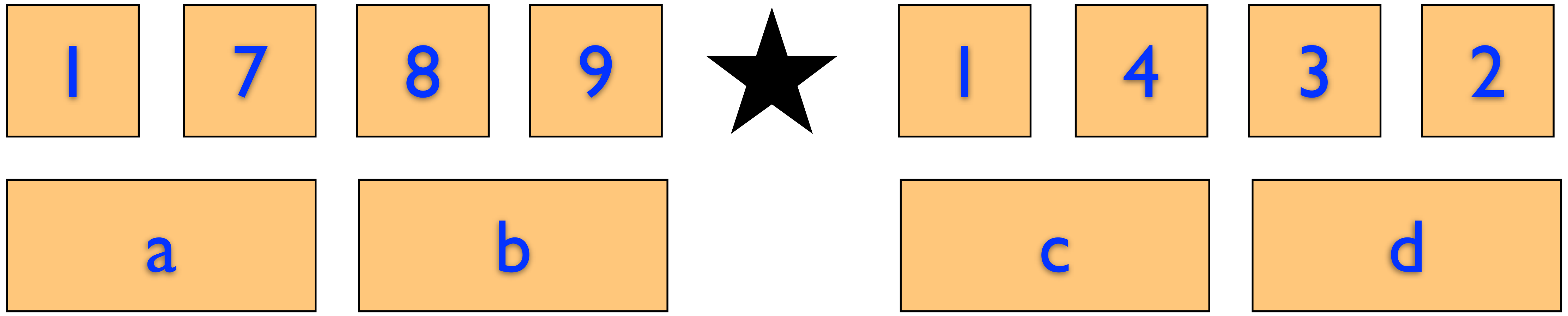
$$O(n^{\log_2(3)}) \sim O(n^{1.589})$$



y
5·10^5
0
250
500
750
x
textbook
fast
1000



$$T(n) = 3T(n/2) + 6O(n)$$



$$T(n) = 3T(n/2) + 6O(n)$$

$$T(n) = 4T(n/2) + 3O(n)$$

simpler proof technique?

❶ classic

induction redux

goal: prove that some property P(k) is true for all k

$\forall k, P(k)$ holds

❶ classic

one long proof...

goal: prove that some property P(k) is true for all k

$\forall k, P(k)$ holds

❶ induction redux

classic
base case: $P(1)$

classic
inductive
step:

$$P(1) \quad \cdots \quad P(k)$$ true implies $P(k+1)$ true

❷

induction redux asymptotic style

base case: $P(n^*)$

inductive step: $P(n^*)$ $\ldots$ $P(k)$ true implies $P(k+1)$ true

simpler proof

(guess +chk)

$$T(n) = 3T(n/2) + 6O(n)$$

$$T(n) = 3T(n/2) + 6O(n)$$ (guess +chk)

want to show: $T(n) = O(n^{\log 3})$

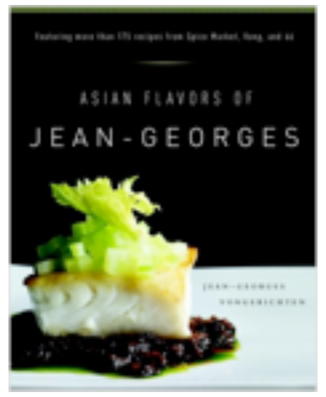
property: $T(n) < n^{\log_2 3} - d'n$

base case: (handled by constants d’ and d”)

inductive step:

simpler proof

?-✓



http://www.drblank.com/law301.jpg

merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge (A, p, q, r)
  ...

```
MERGE(A[1 .. n], m):
  i ← 1; j ← m + 1
  for k ← 1 to n
      if j > n
          B[k] ← A[i]; i ← i + 1
      else if i > m
          B[k] ← A[j]; j ← j + 1
      else if A[i] < A[j]
          B[k] ← A[i]; i ← i + 1
      else
          B[k] ← A[j]; j ← j + 1
  for k ← 1 to n
      A[k] ← B[k]
```

jeff erickson

$$T(n) = 2T(n/2) + n$$

prove:

$$T(n) = 2T(n/2) + n$$

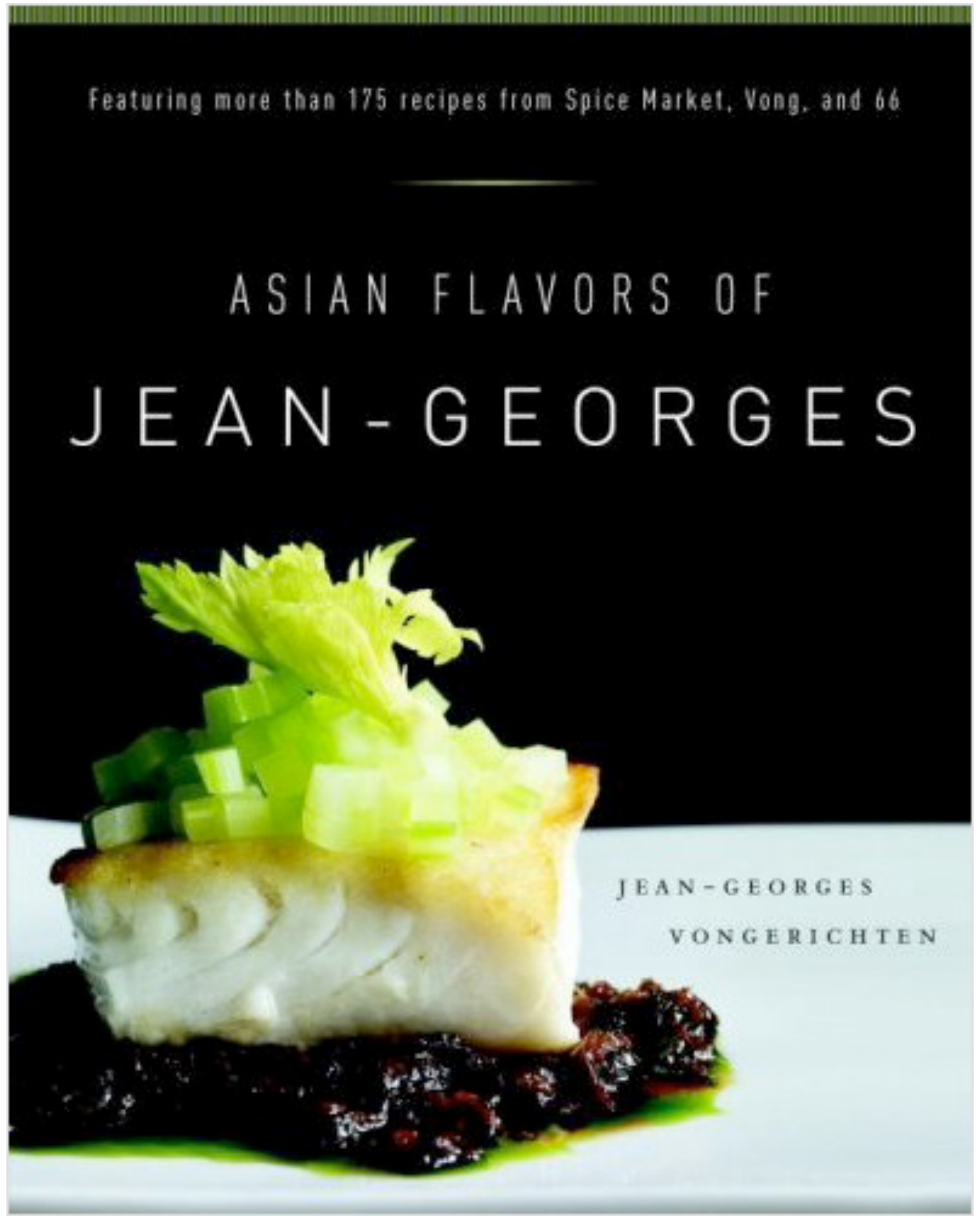
prove:

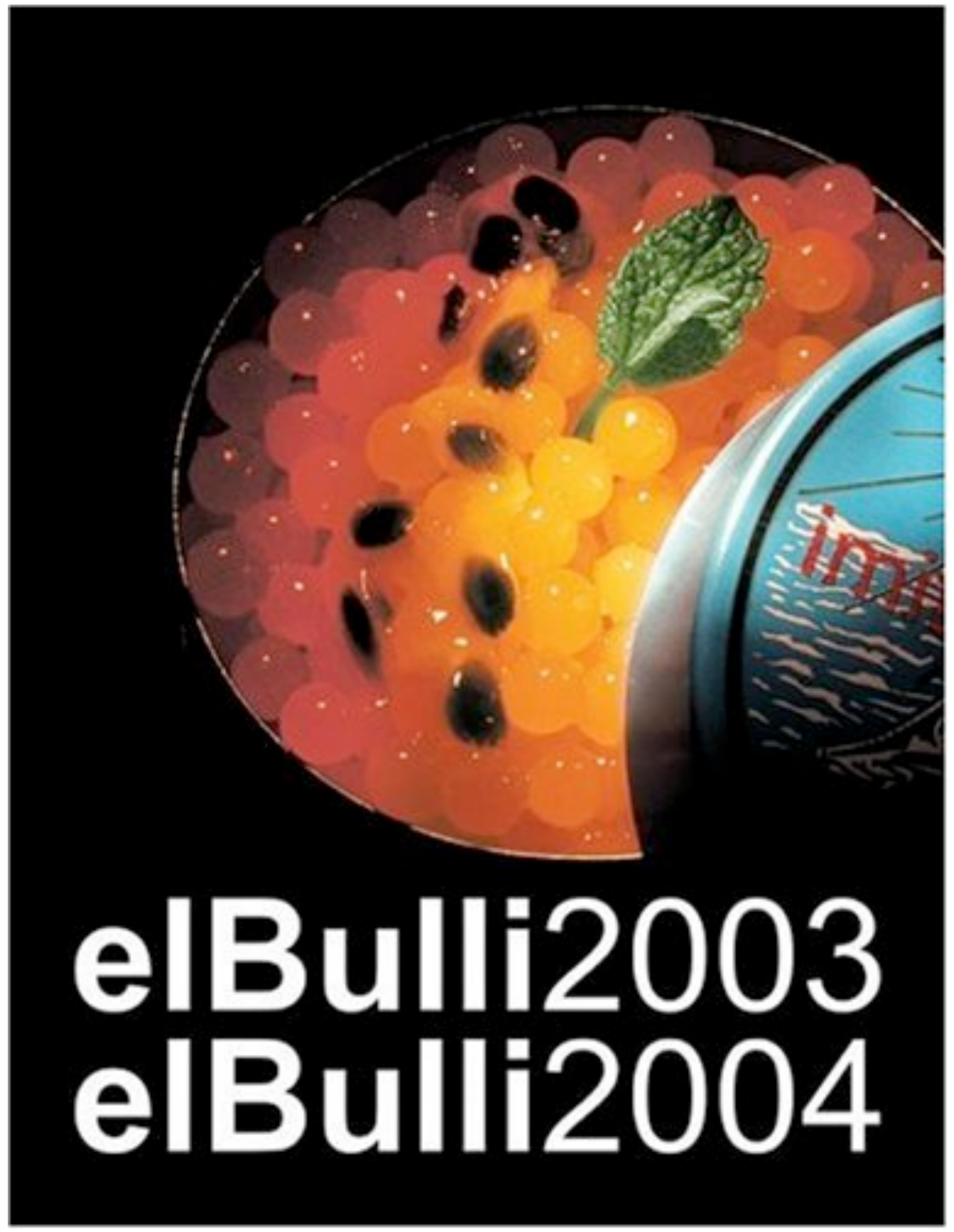
hypothesis:

base case:

inductive step:

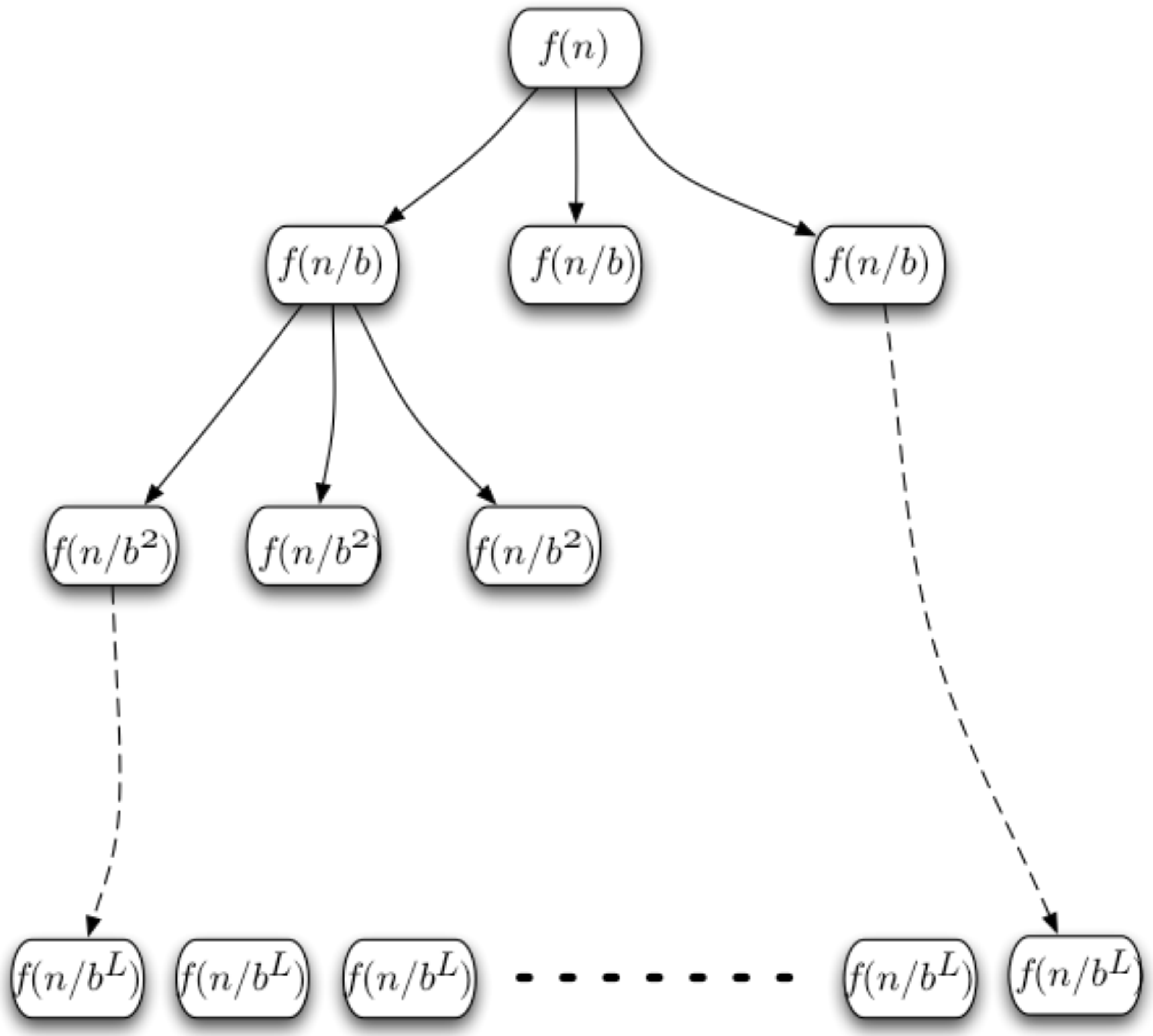
cookbook


Featuring more than 175 recipes from Spice Market, Vong, and 66
ASIAN FLAVORS OF
JEAN-GEORGES
JEAN-GEORGES
VONGERICHTEN


elBulli2003
elBulli2004

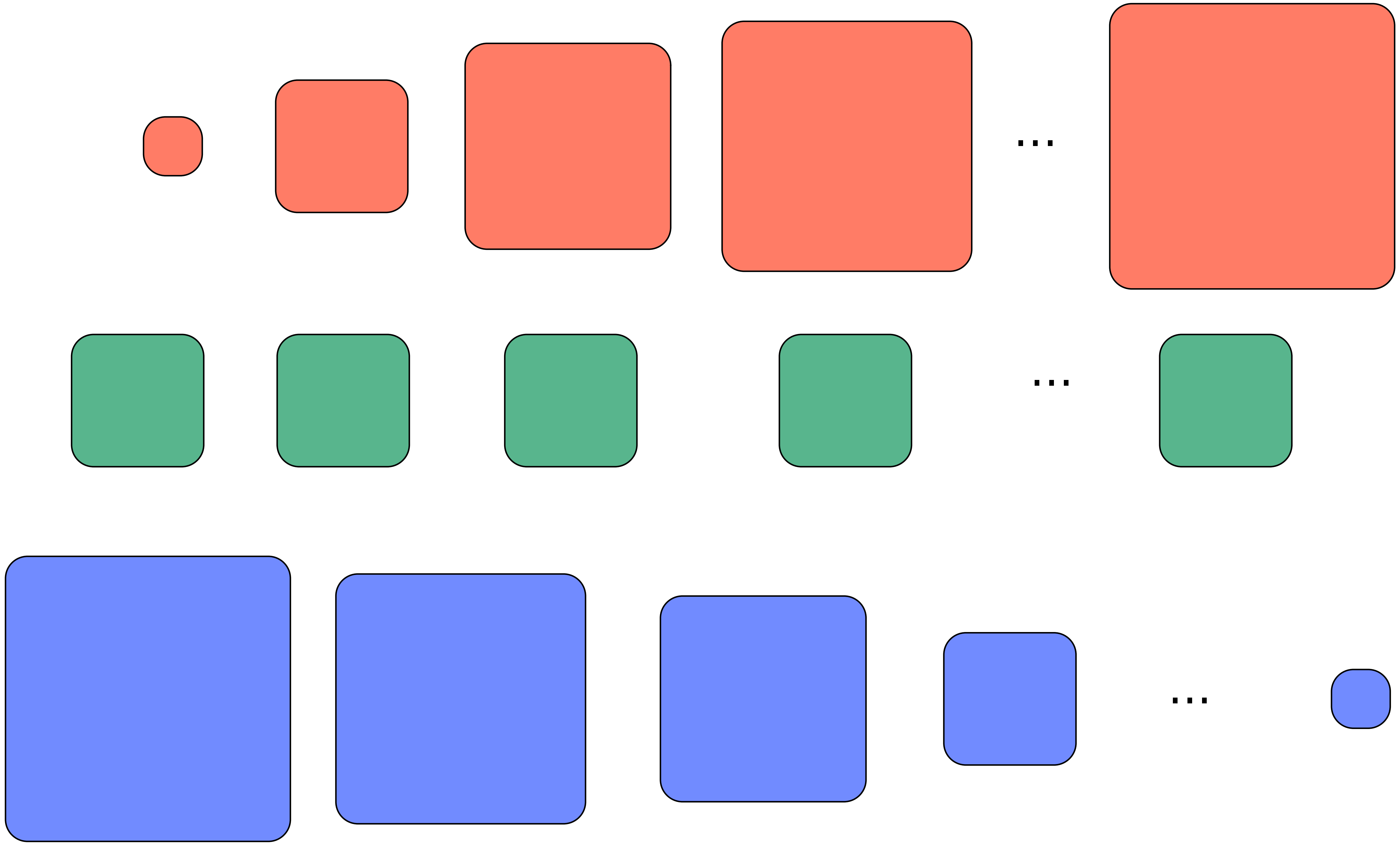
$$T(n) = aT(n/b) + f(n)$$

$$T(n) = aT(n/b) + f(n)$$



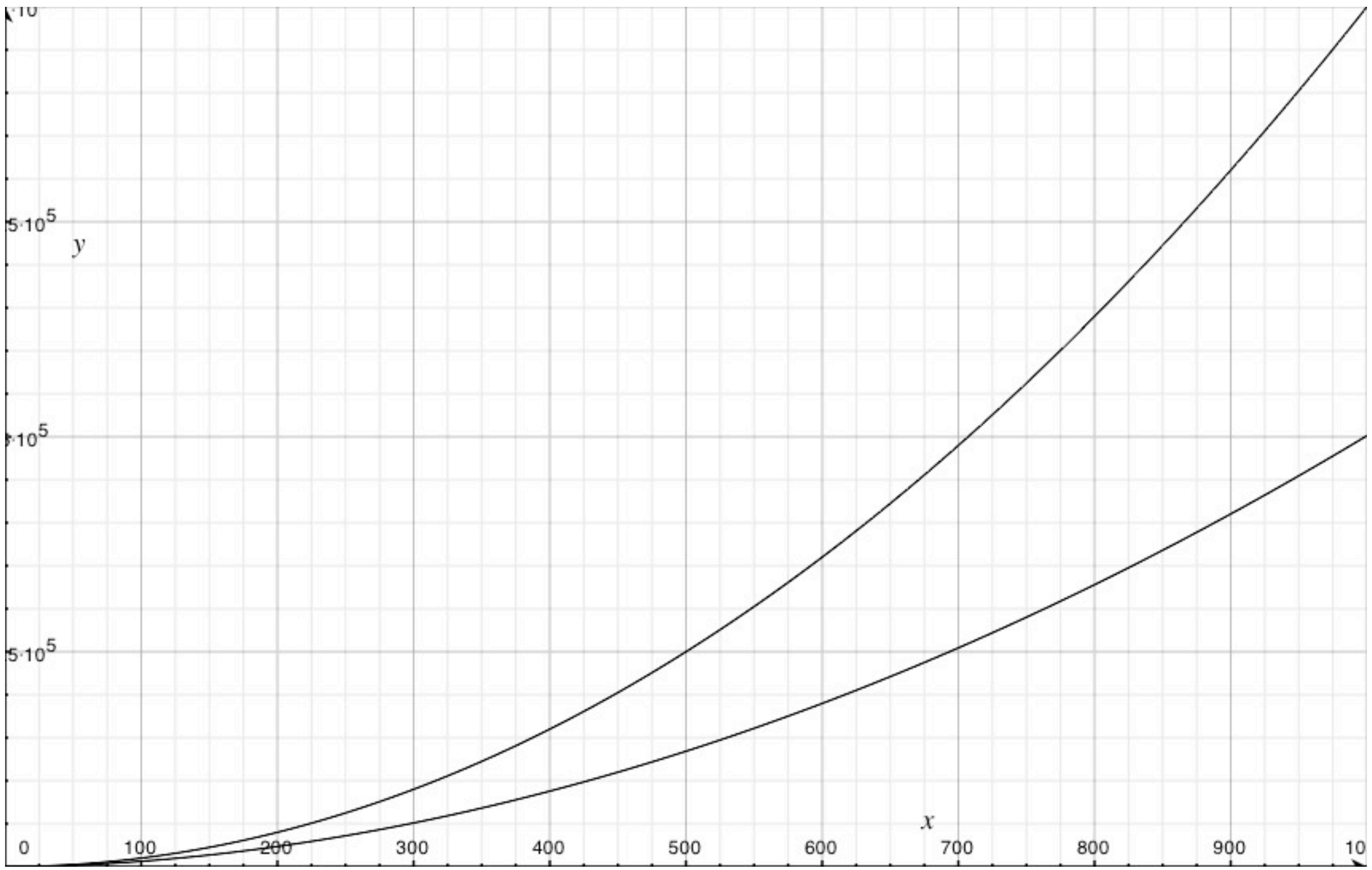
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$



$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: $$f(n) = O(n^{\log_b a - \epsilon})$$



y
x
5·10^5
0
100
200
300
400
500
600
700
800
900

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: $f(n) = \Theta(n^{\log_b a - \epsilon})$

example: $T(n) = 4T(n/2) + n$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1 (cont):