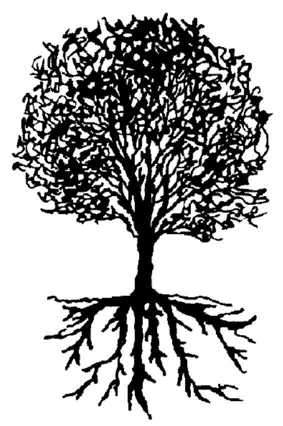
LA 5800

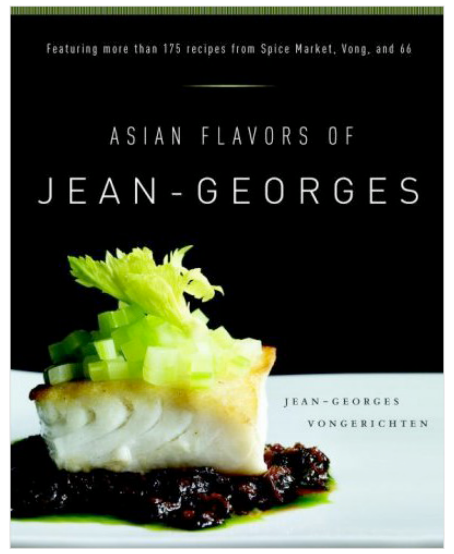
jan 28/31 2022

shelat

Announcements on H1

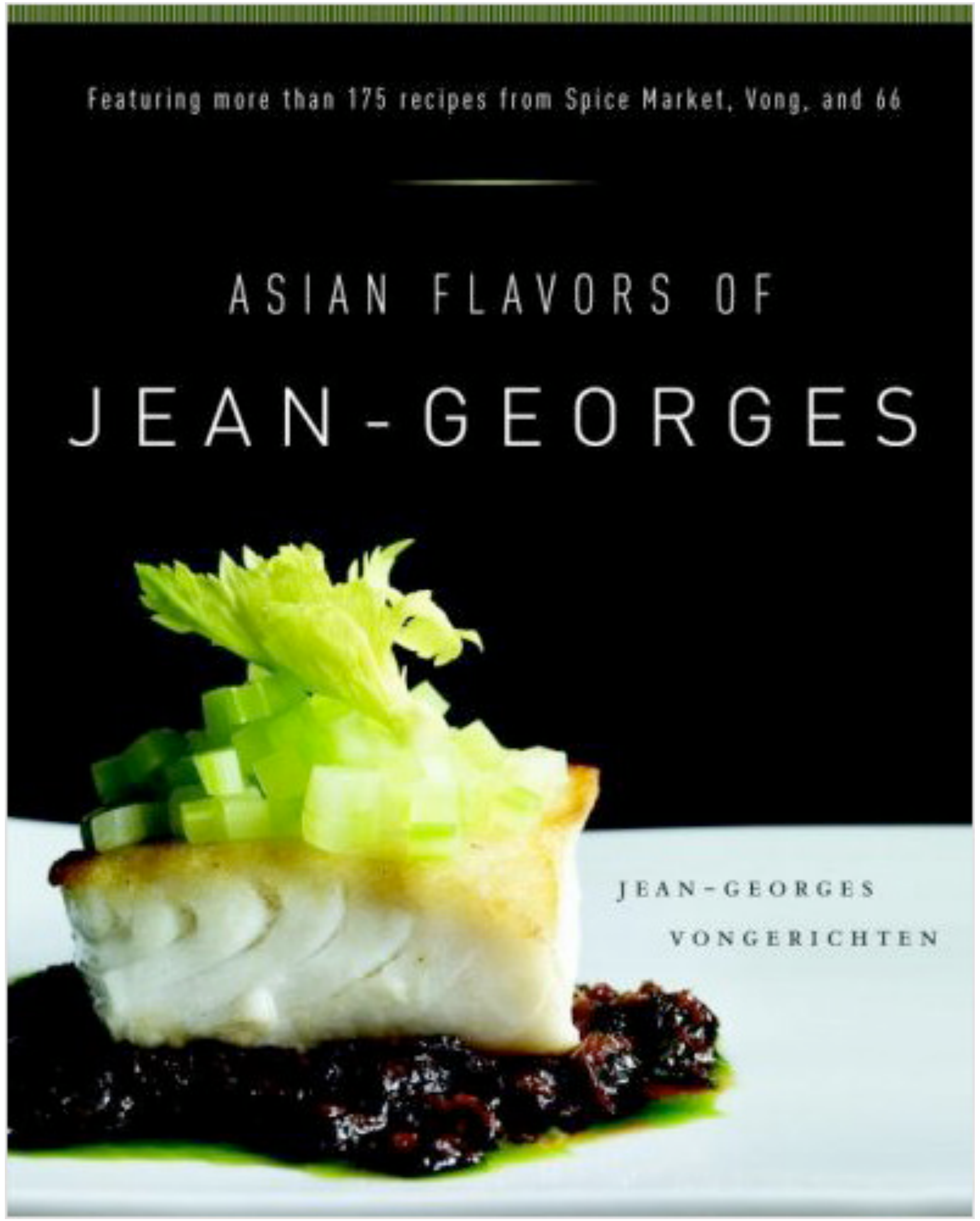


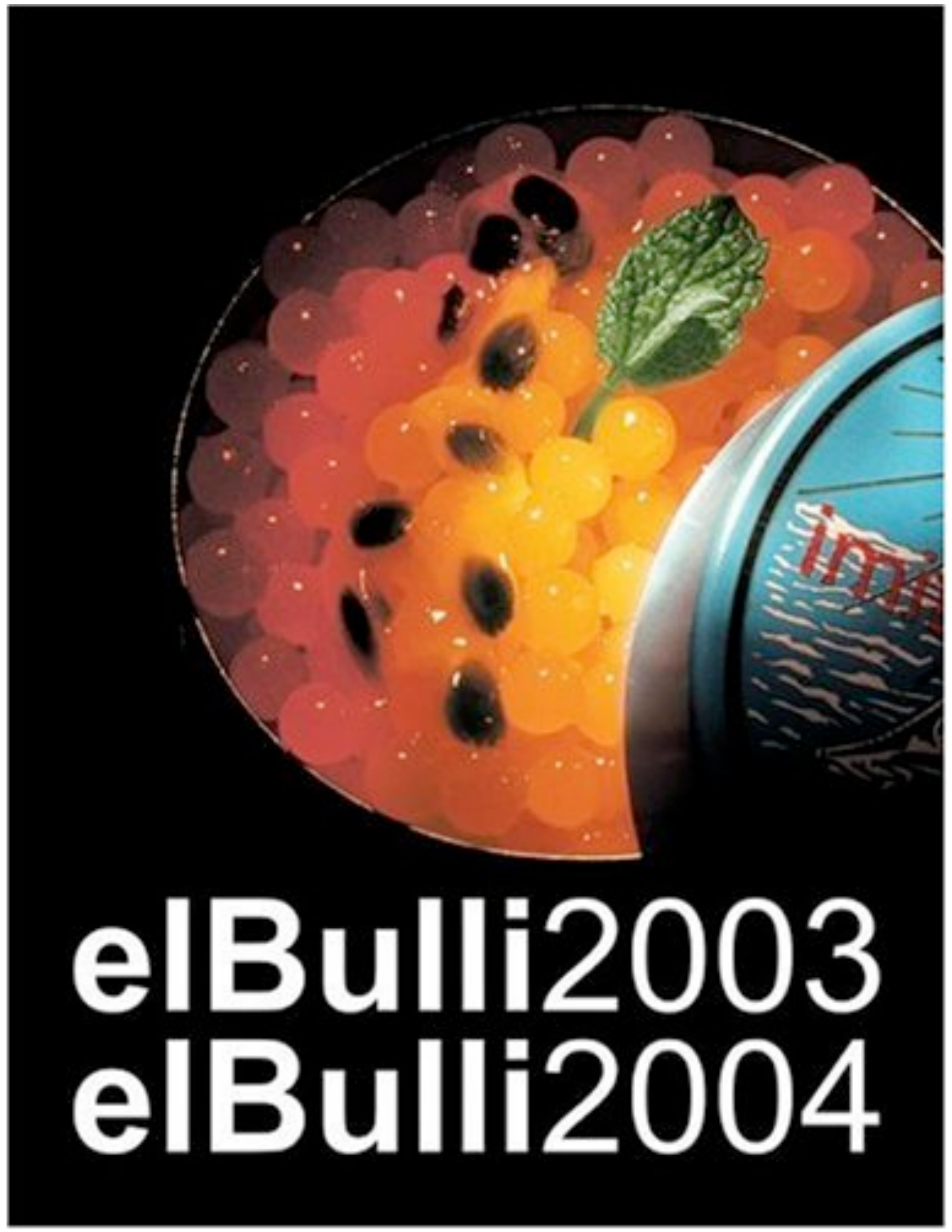
?-✓



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

cookbook

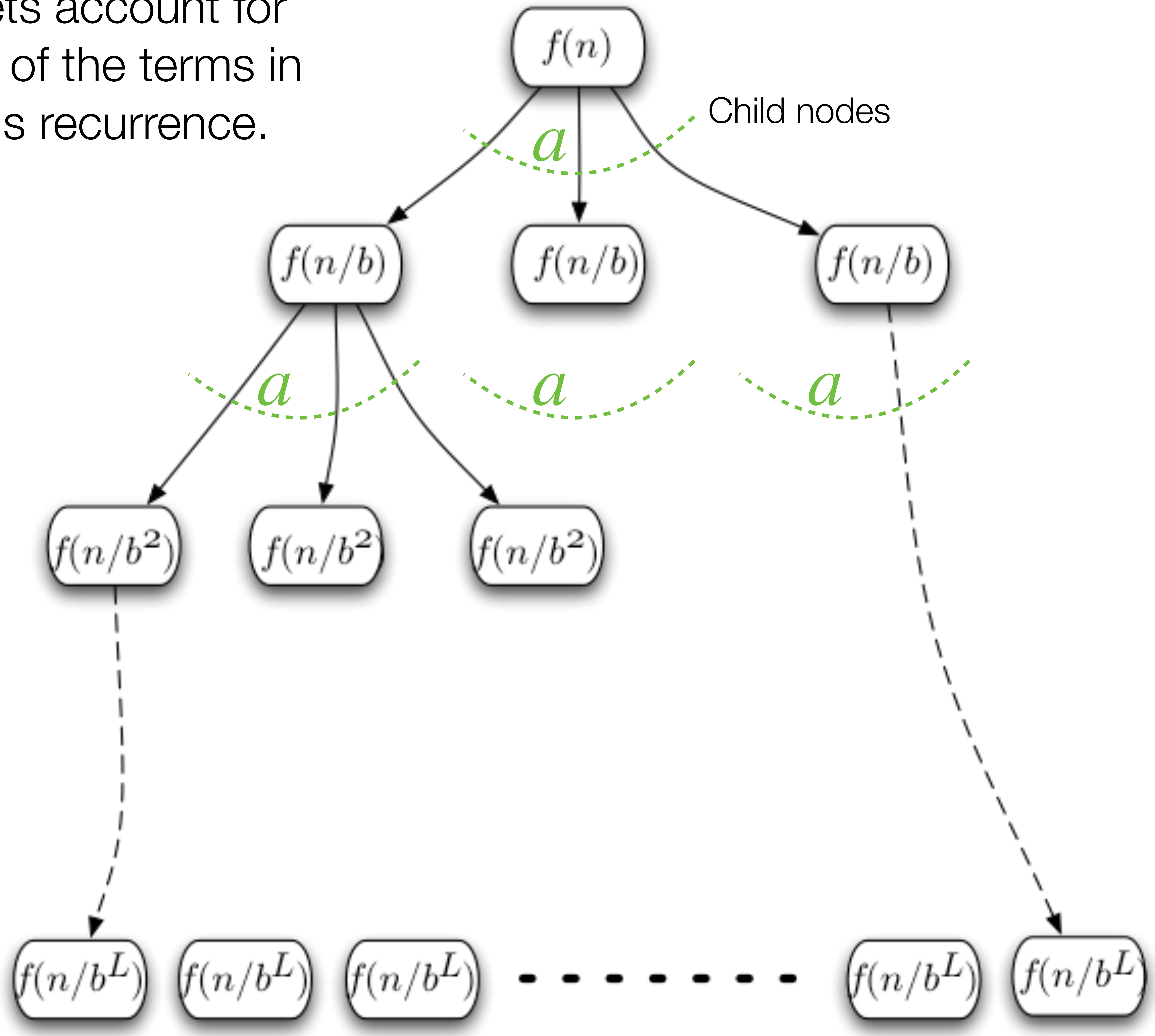

Featuring more than 175 recipes from Spice Market, Vong, and 66
ASIAN FLAVORS OF
JEAN-GEORGES
JEAN-GEORGES
VONGERICHTEN


elBulli2003
elBulli2004

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

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

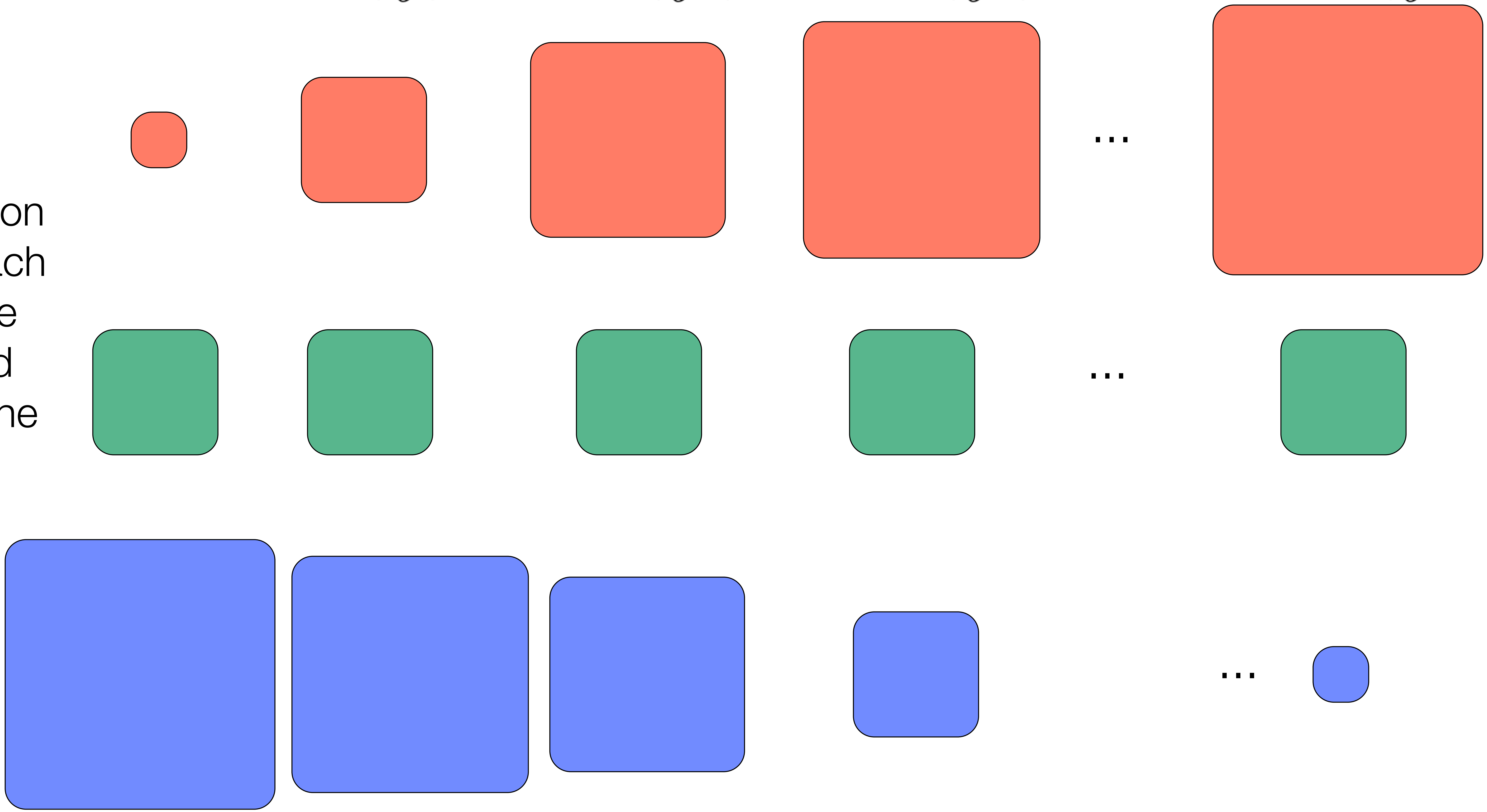
Lets account for all of the terms in this recurrence.



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

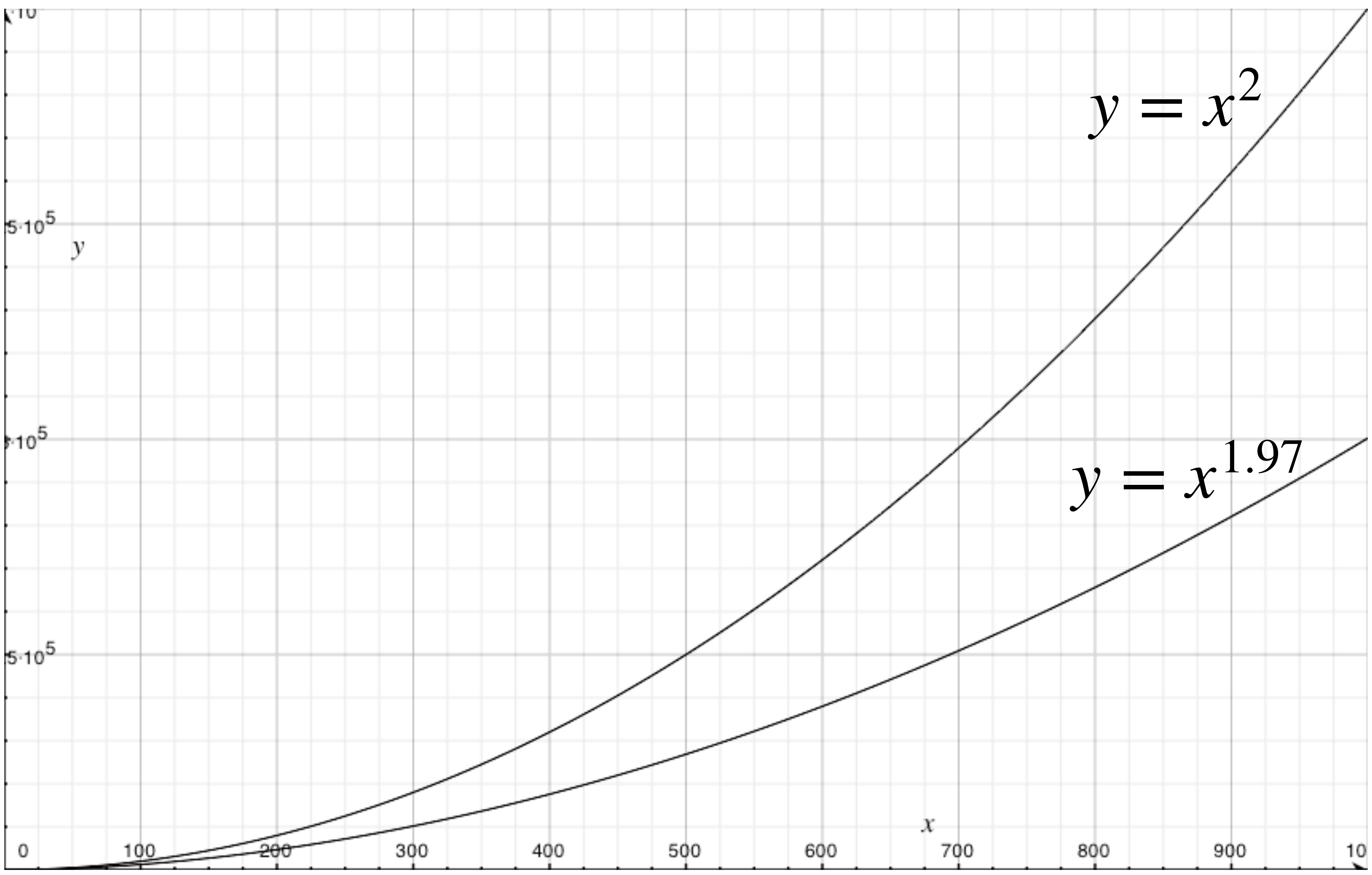
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

Comparison of how each term in the sum could relate to the others



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

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



$y = x^2$
$y = x^{1.97}$
y
x
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: $$f(n) = O(n^{\log_b a - \epsilon})$$

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

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

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

$$T(n) \leq cn^{\log_b a - \epsilon} + ac\left(\frac{n}{b}\right)^{\log_b a - \epsilon} + a^2 c\left(\frac{n}{b^2}\right)^{\log_b a - \epsilon} + \cdots + a^L c\left(\frac{n}{b^L}\right)^{\log_b a - \epsilon}$$

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

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

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + \left(\frac{a}{b^{\log_b a - \epsilon}}\right) + \left(\frac{a^2}{b^{2(\log_b a - \epsilon)}}\right) + \cdots + \left(\frac{a^L}{b^{L(\log_b a - \epsilon)}}\right)\right]$$

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

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

$$T(n) \le cn^{\log_b a-\epsilon} + ac\left(\frac{n}{b}\right)^{\log_b a-\epsilon} + a^2c\left(\frac{n}{b^2}\right)^{\log_b a-\epsilon} + \cdots + a^Lc\left(\frac{n}{b^L}\right)^{\log_b a-\epsilon}$$

$$= cn^{\log_b a-\epsilon}\left[1 + \left(\frac{a}{b^{\log_b a-\epsilon}}\right) + \left(\frac{a^2}{b^{2(\log_b a-\epsilon)}}\right) + \cdots + \left(\frac{a^L}{b^{L(\log_b a-\epsilon)}}\right)\right]$$

$$= cn^{\log_b a-\epsilon}\left[1 + \left(\frac{a}{a/b^{\epsilon}}\right) + \left(\frac{a^2}{a^2/b^{2\epsilon}}\right) + \cdots + \left(\frac{a^L}{a^L/b^{L\epsilon}}\right)\right]$$

$$= cn^{\log_b a-\epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

Since $b > 1, \epsilon > 0$

then $b^{\epsilon} > 1$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon}n^{\epsilon} - 1}{b^{\epsilon} - 1}\right]$$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon}n^{\epsilon} - 1}{b^{\epsilon} - 1}\right]$$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$\leq \left[\frac{cb^{\epsilon}}{b^{\epsilon} - 1}\right] n^{\log_b a - \epsilon} n^{\epsilon} = O(n^{\log_b a})$$

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

case 1: Lower bound

We have:

$$T(n) \geq a^L f\left(\frac{n}{b^L}\right)$$

$$\geq n^{\log_b a}$$

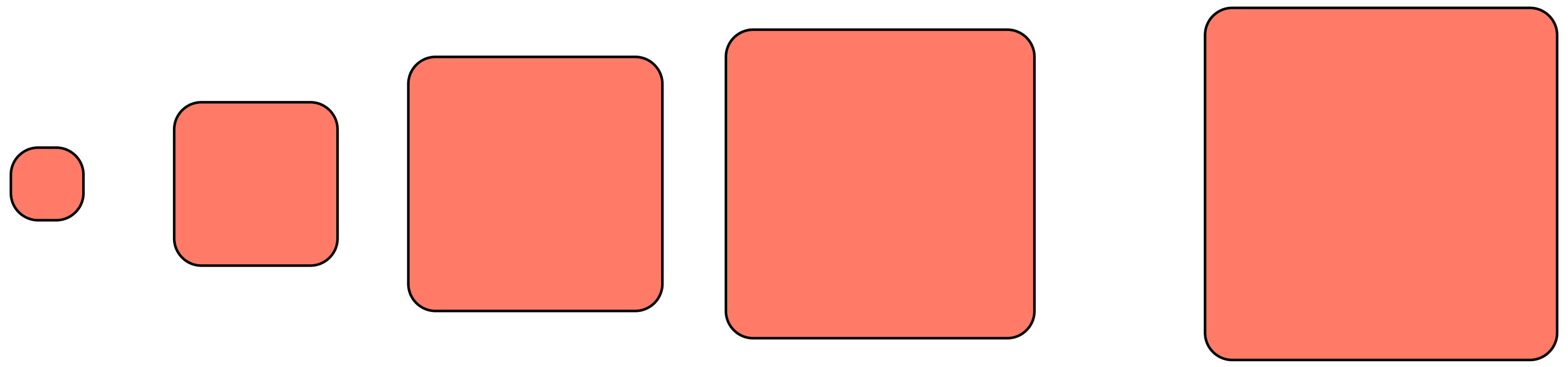
Recall that L is the depth of the recursion and

$$a^L = a^{\log_b n}$$

Master's Theorem

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

Case 1



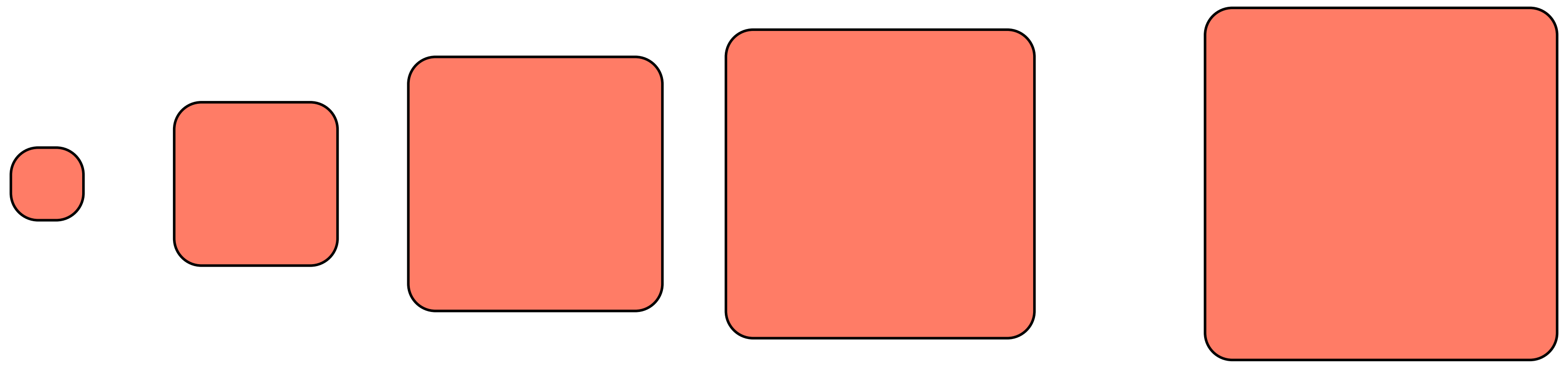
$$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$$

Then $T(n) = \Theta(n^{\log_b a})$

Master's Theorem

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

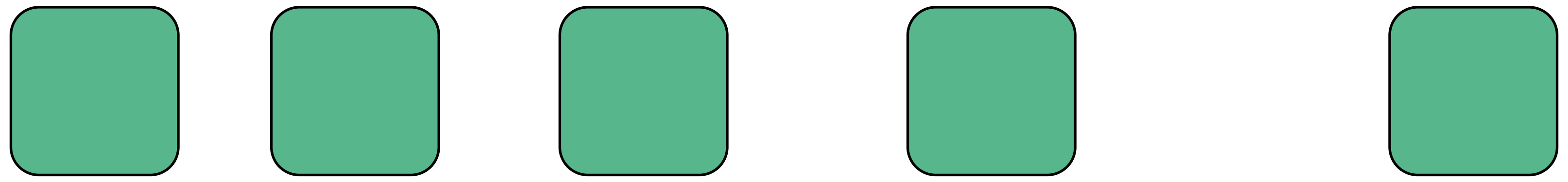
Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

Then $T(n) = \Theta(n^{\log_b a})$

Case 2

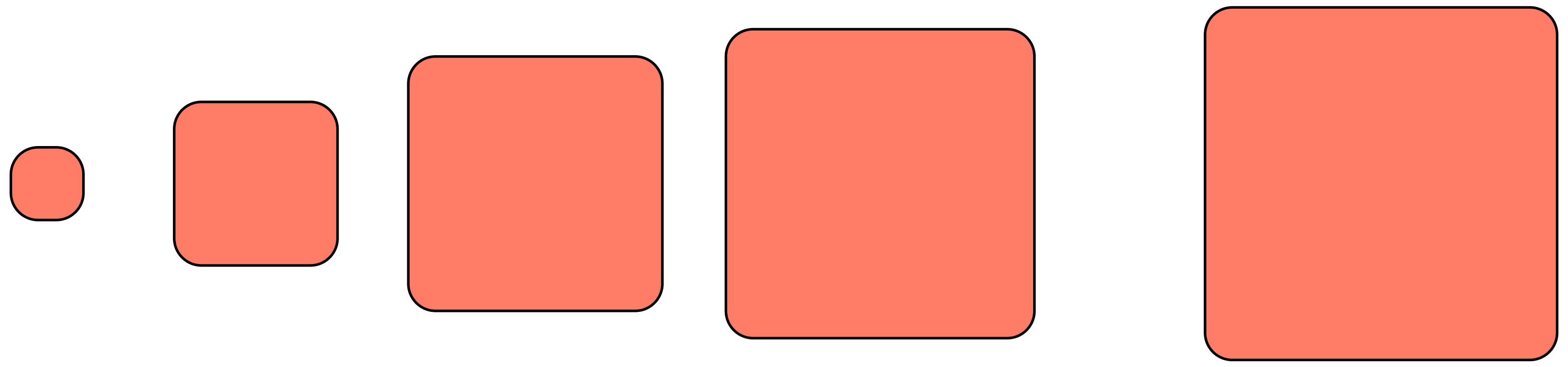


$f(n) = \Theta(n^{\log_b a})$

Master's Theorem

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

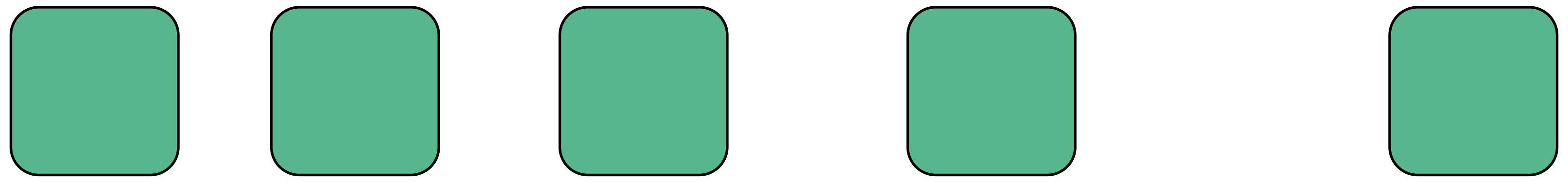
Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

Then $T(n) = \Theta(n^{\log_b a})$

Case 2



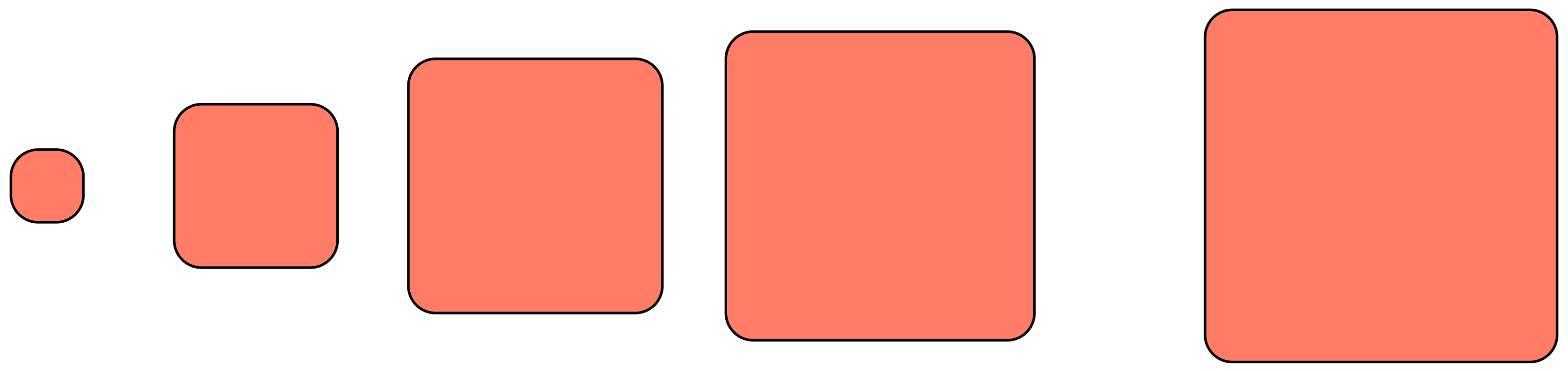
$f(n) = \Theta(n^{\log_b a})$

Then $T(n) = \Theta(n^{\log_b a} \log n)$

Master's Theorem

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

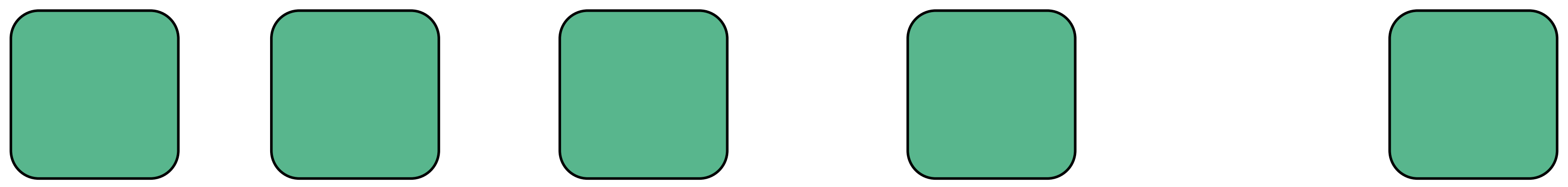
Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

Then $T(n) = \Theta(n^{\log_b a})$

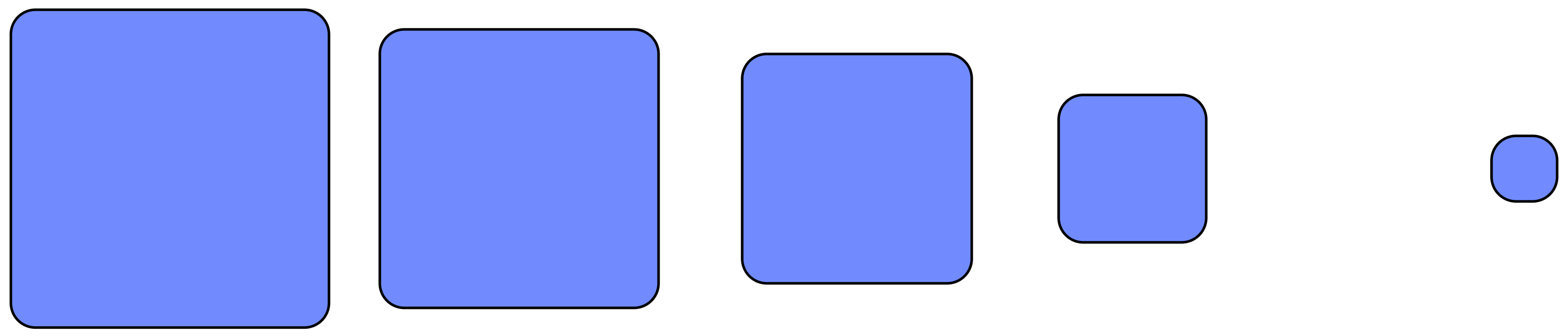
Case 2



$f(n) = \Theta(n^{\log_b a})$

Then $T(n) = \Theta(n^{\log_b a} \log n)$

Case 3



$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$ and $\exists c < 1,\ af(n/b) < cf(n)$

Then $T(n) = \Theta(f(n))$

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

case 2: $c' n^{\log_b a} < f(n) < c n^{\log_b a}$

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

case 2: $c' n^{\log_b a} < f(n) < c n^{\log_b a}$

$$T(n) < c n^{\log_b a}\left[1 + \left(\frac{a}{b^{\log_b a}}\right) + \left(\frac{a^2}{b^{2\log_b a}}\right) + \cdots + \left(\frac{a^L}{b^{L\log_b a}}\right)\right]$$

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

case 2: $c' n^{\log_b a} < f(n) < c n^{\log_b a}$

$$T(n) < c n^{\log_b a}\left[1 + \left(\frac{a}{b^{\log_b a}}\right) + \left(\frac{a^2}{b^{2\log_b a}}\right) + \cdots + \left(\frac{a^L}{b^{L\log_b a}}\right)\right]$$

$$= c n^{\log_b a}\left[1 + 1 + \cdots 1\right]$$

$$= c n^{\log_b a}\left[\log_b n\right] = O(n^{\log_b a} \log n)$$

Similar argument for lower bound.

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$af\left(\frac{n}{b}\right) < cf(n)$$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$af\left(\frac{n}{b}\right) < cf(n)$$

$$a^2 f\left(\frac{n}{b^2}\right) = a\left[af\left(\frac{n}{b^2}\right)\right] < a\left[cf\left(\frac{n}{b}\right)\right] = c\left[af\left(\frac{n}{b}\right)\right] < c^2 f(n)$$

$$a^3 f\left(\frac{n}{b^3}\right) < c \cdot a^2 f\left(\frac{n}{b^2}\right) < c^3 f(n)$$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$T(n) < f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$$

$$= f(n)\left[1 + c + c^2 + \cdots c^L\right]$$

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

It is important that c<1 for the sum term to be bounded by a constant

Similar argument for lower bound.

example from last class: $T(n) = 8T(n/2) + \Theta(n^2)$

example from last class: $T(n) = 8T(n/2) + \Theta(n^2)$

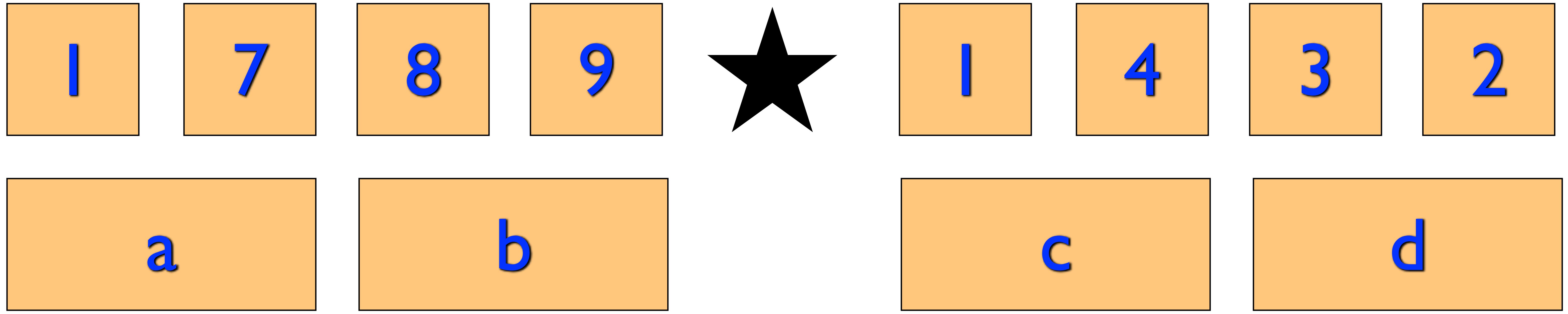
$$a = 8, b = 2, f(n) = \Theta(n^2)$$

example from last class: $T(n) = 8T(n/2) + \Theta(n^2)$

$$a = 8, b = 2, f(n) = \Theta(n^2)$$

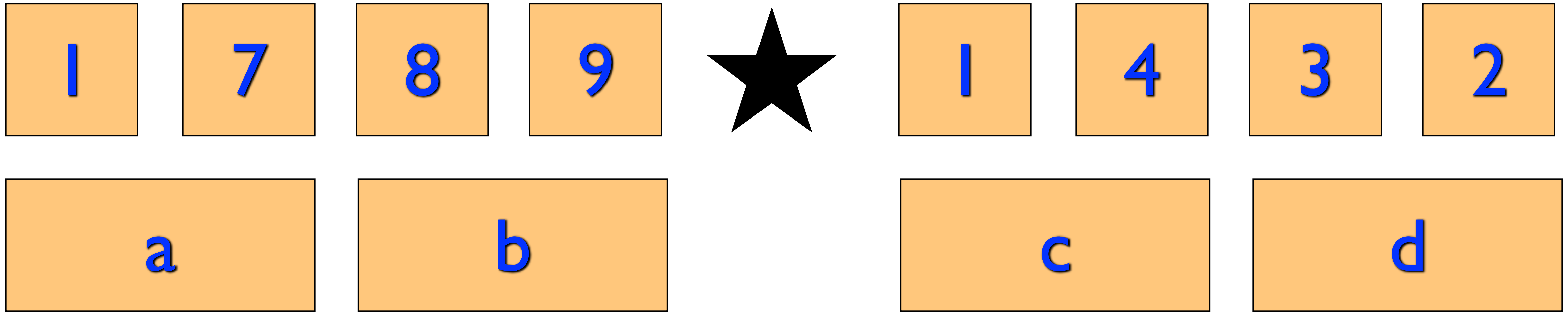
Since $f(n) < cn^2 = O(n^{\log_2 8} - 0.1) = O(n^{2.9})$ then Case 1 applies.

Therefore $T(n) = \Theta(n^{\log_2 8}) = \Theta(n^3)$



Schoolbook approach

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



Schoolbook approach

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

$$a = 4, b = 2, f(n) = O(n)$$

Therefore, case1,

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

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

Since $24 = \Theta(n^{\log_{17/14} 1}) = \Theta(n^0)$, case 2 applies.

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

Since $24 = \Theta(n^{\log_{17/14} 1}) = \Theta(n^0)$, case 2 applies.

Therefore $T(n) = \Theta(\log n)$

$$T(n) = 2T(n/2) + n^3$$

$$T(n) = 2T(n/2) + n^3$$

Since $n^3 = \Omega(n^{\log_2 2+\epsilon})$ and $2\left(\frac{n}{2}\right)^3 < \left(\frac{1}{2}\right)n^3$ Case 3 applies.

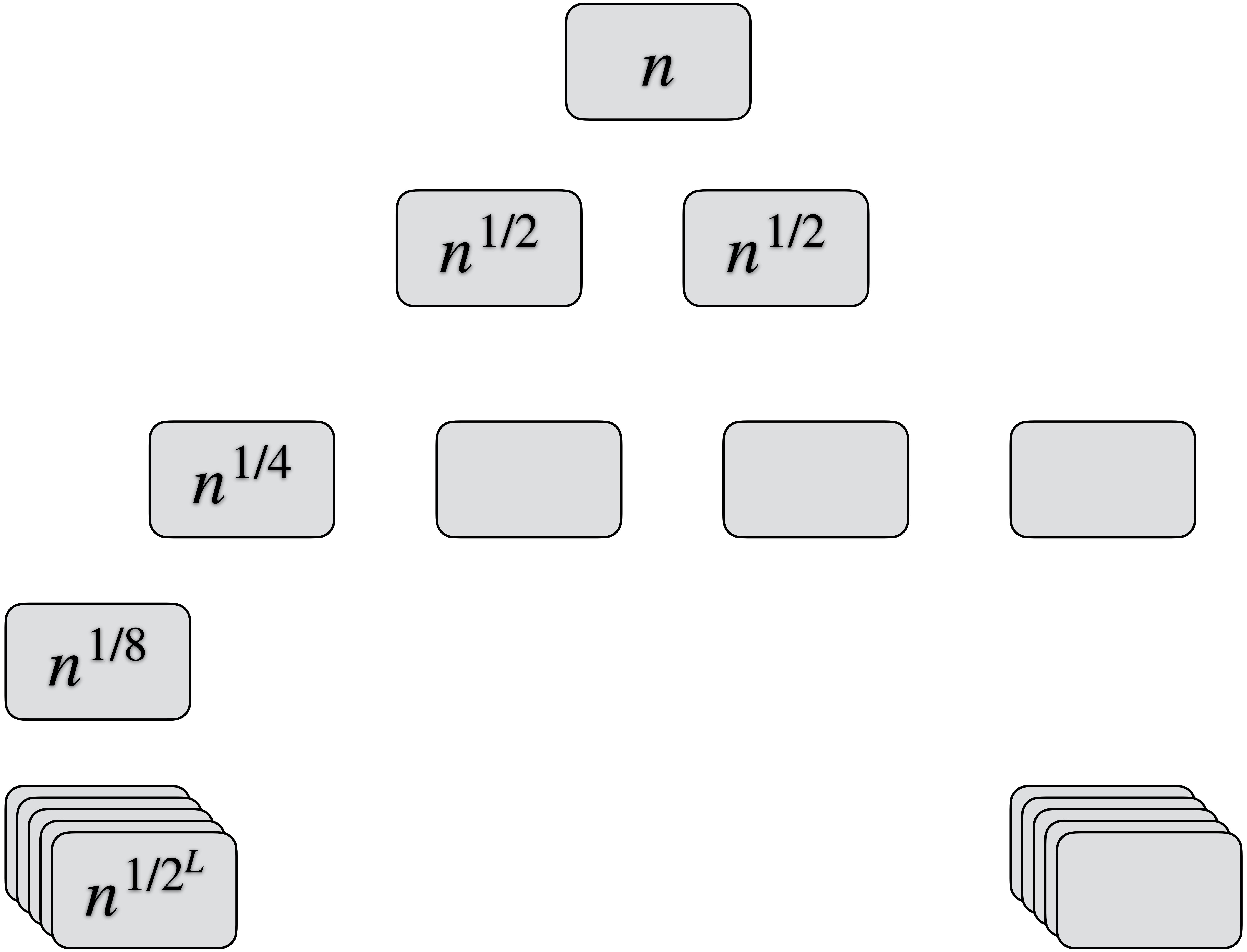
$$T(n) = 16T(n/4) + n^2$$

$$T(n) = 7T(n/2) + \Theta(n^2)$$


8
13
FIFA

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$



$$\lg n$$

$$2 \lg(n^{1/2})$$

$$2^2 \lg(n^{1/2^2})$$

$$2^3 \lg(n^{1/2^3})$$

$$2^L \lg(n^{1/2^L})$$

How to solve for L?

$$n^{\frac{1}{2^L}} = 2$$

Take logs on both sides:

$$\frac{1}{2^L} \log n = \log(2)$$

How to solve for L?

$$n^{\frac{1}{2^L}} = 2$$

Take logs on both sides:

$$\frac{1}{2^L} \log n = \log(2)$$

Then multiply both sides by 2^L, and take logs again.

$$\log \log n = L$$

How to solve for L?

$$n^{\frac{1}{2^L}} = 2$$

For our purposes, this value can be a constant. Why not 1?

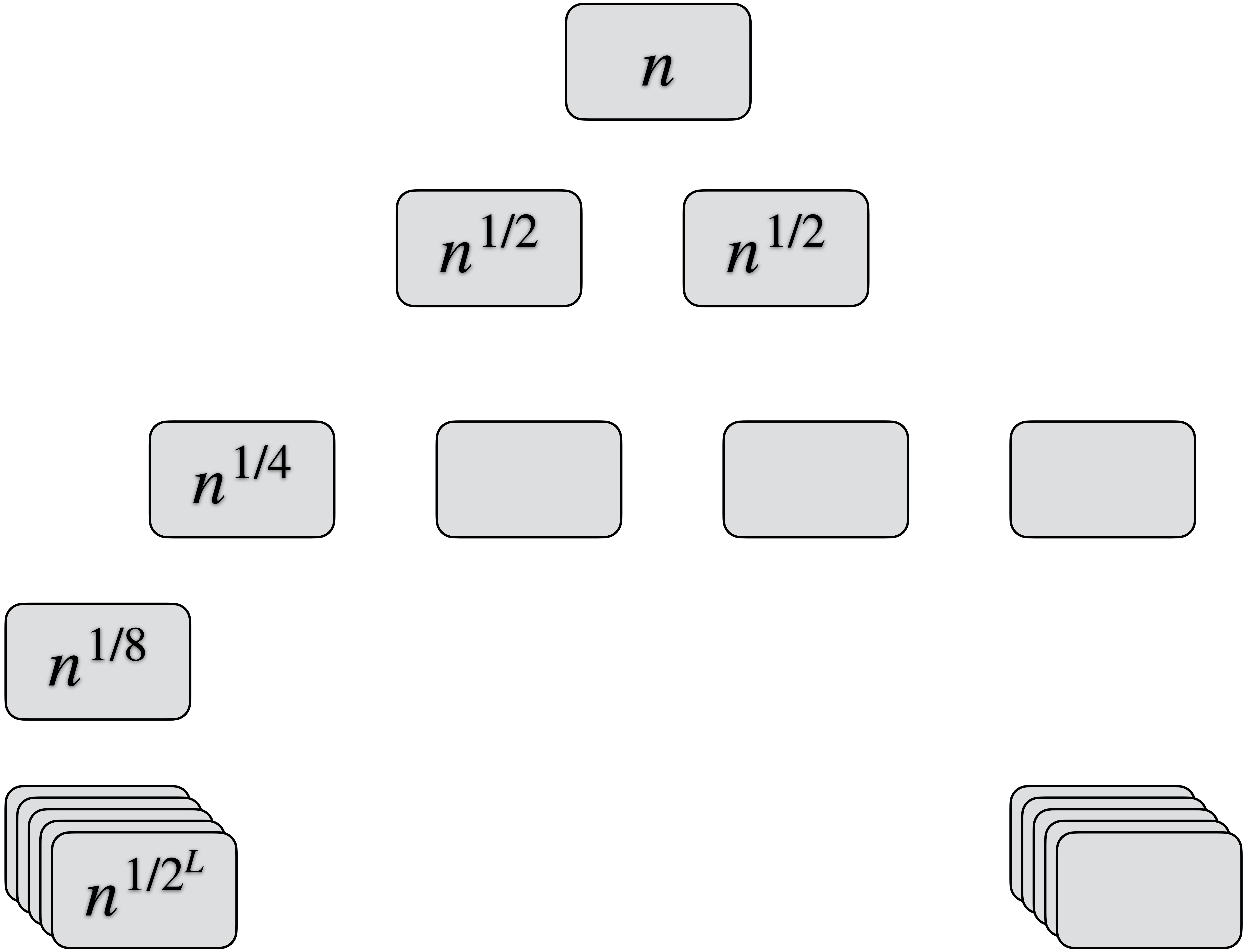
Take logs on both sides:

$$\frac{1}{2^L} \log n = \log(2)$$

Then multiply both sides by 2^L, and take logs again.

$$\log \log n = L$$

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$



$$\lg n$$

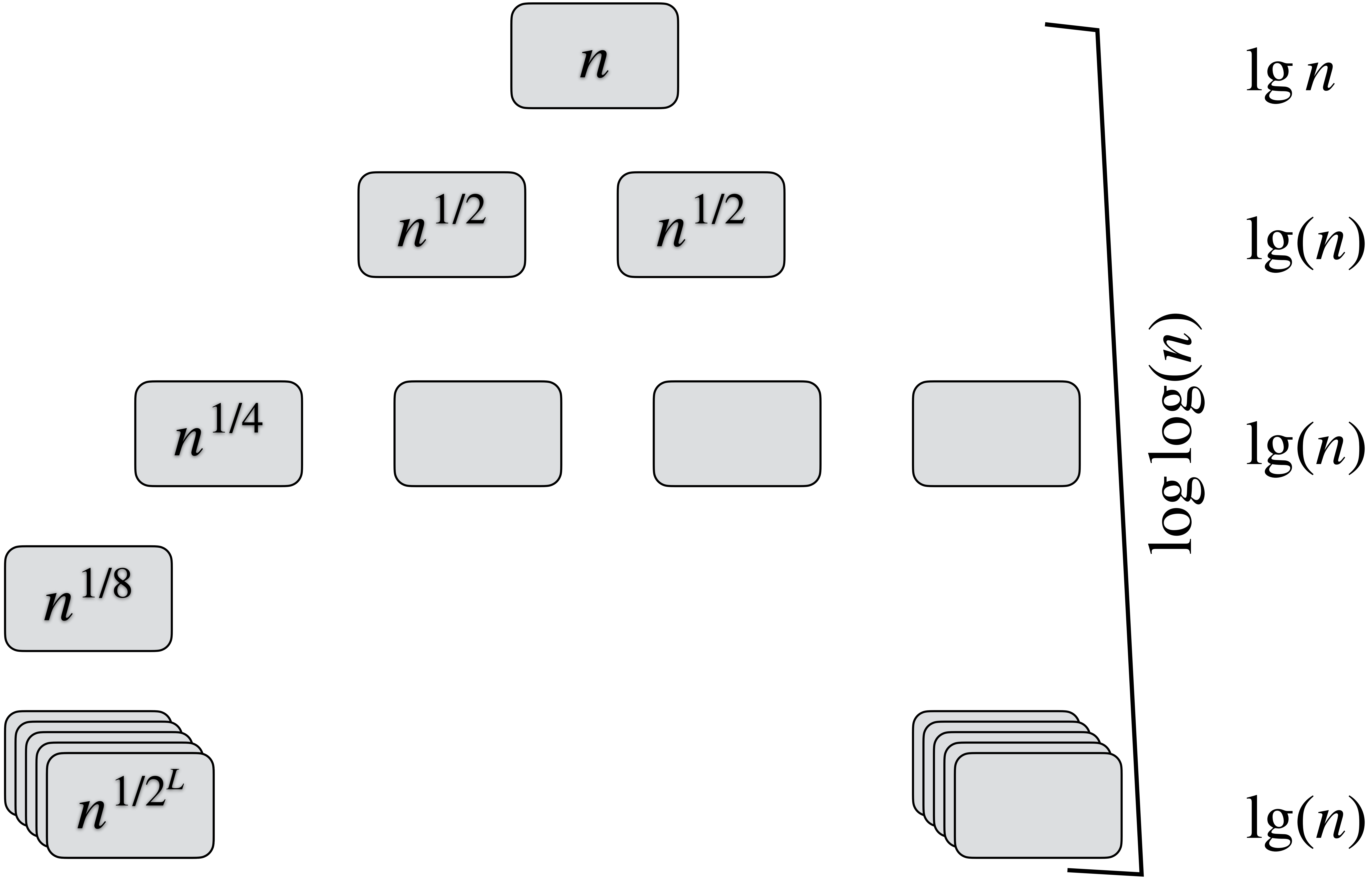
$$2 \lg(n^{1/2})$$

$$2^2 \lg(n^{1/2^2})$$

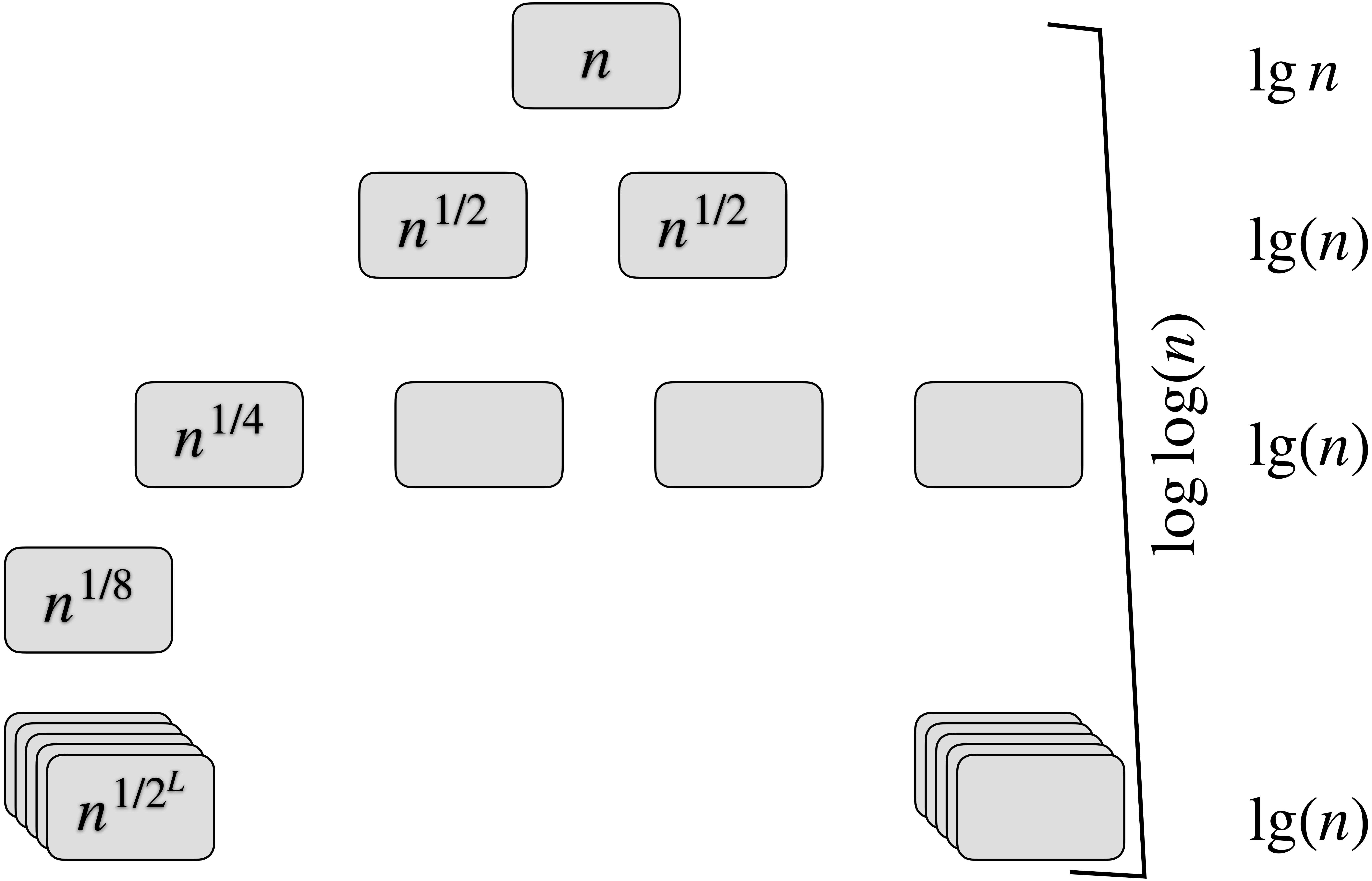
$$2^3 \lg(n^{1/2^3})$$

$$2^L \lg(n^{1/2^L})$$

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$



$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$



Sums to $O(\log n \log\log n)$

Use Induction to prove.

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$

Lets rewrite with $m = \log n$

$T(2^m) = 2T(2^{m/2}) + c \cdot m$

Define $S(m) = T(2^m)$

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$

Lets rewrite with $m = \log n$

$$T(2^m) = 2T(2^{m/2}) + c \cdot m$$

Define $S(m) = T(2^m)$

$$S(m) = 2S(m/2) + \Theta(m)$$

$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$

Lets rewrite with $m = \log n$

$$T(2^m) = 2T(2^{m/2}) + c \cdot m$$

Define $S(m) = T(2^m)$

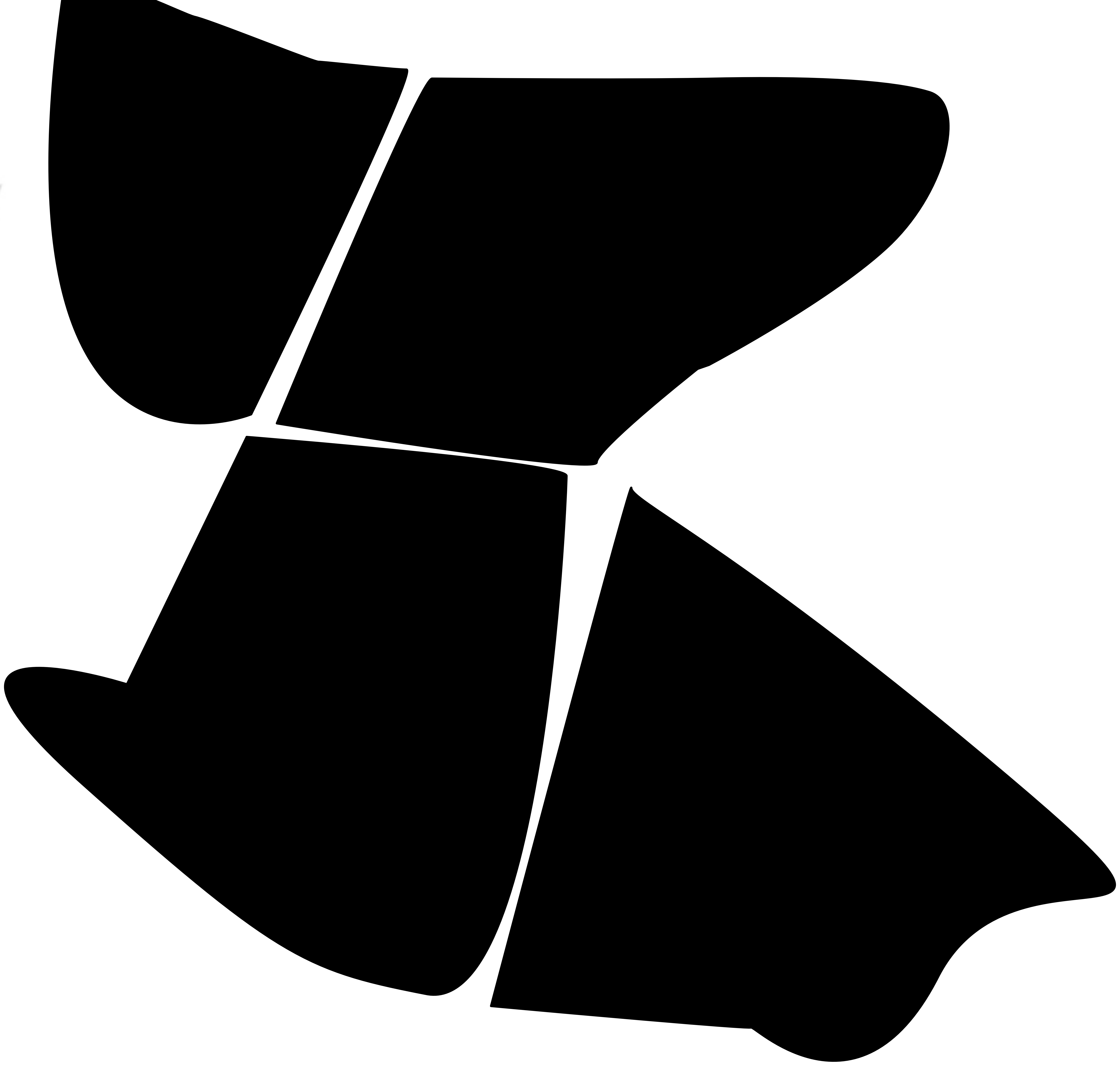
$$S(m) = 2S(m/2) + \Theta(m)$$

Apply Master's Thm case 2: $S(m) = \Theta(m \log m)$

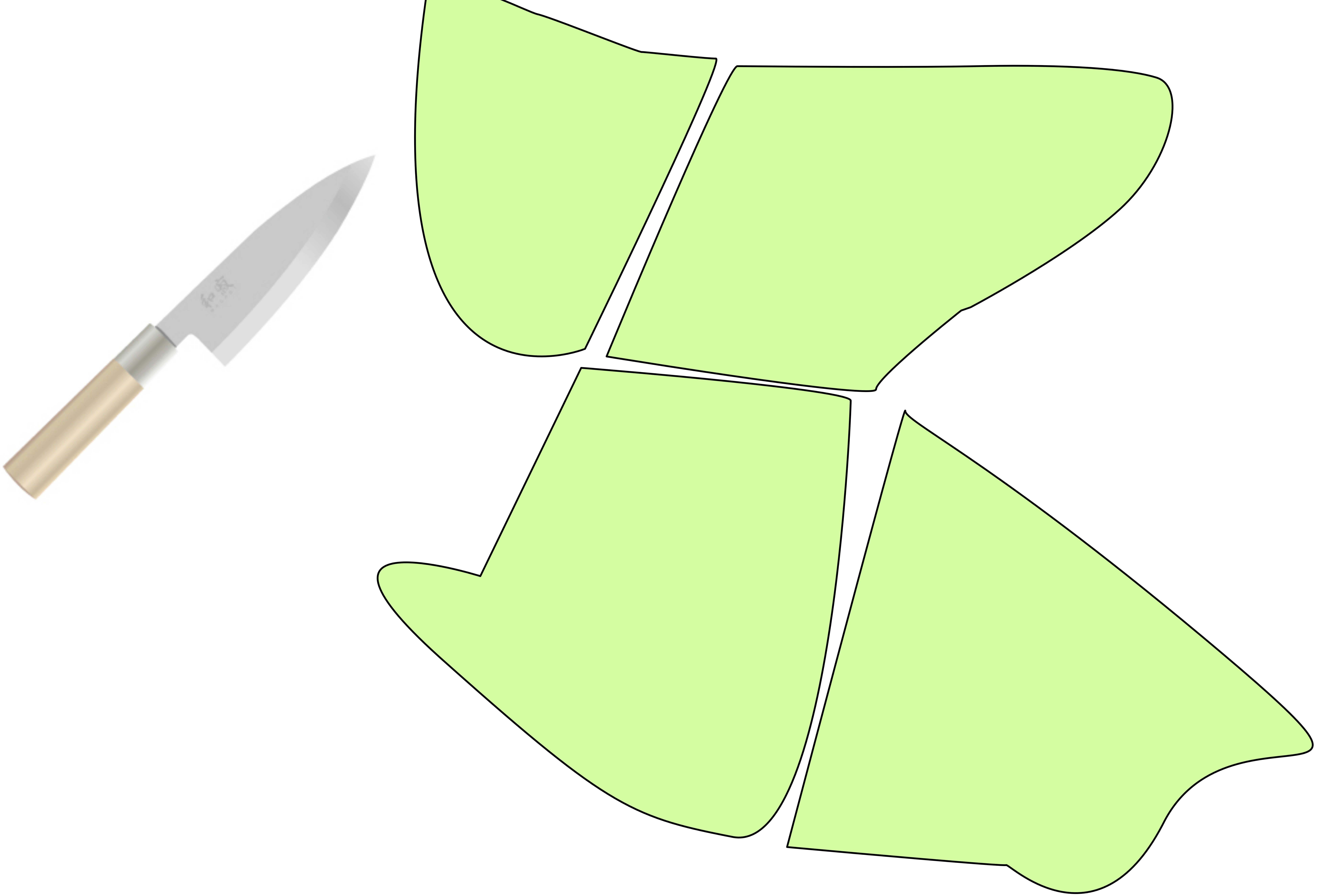
Since $m = \log n$, we have $T(n) = \Theta(\log n \log \log n)$

divide
& conquer

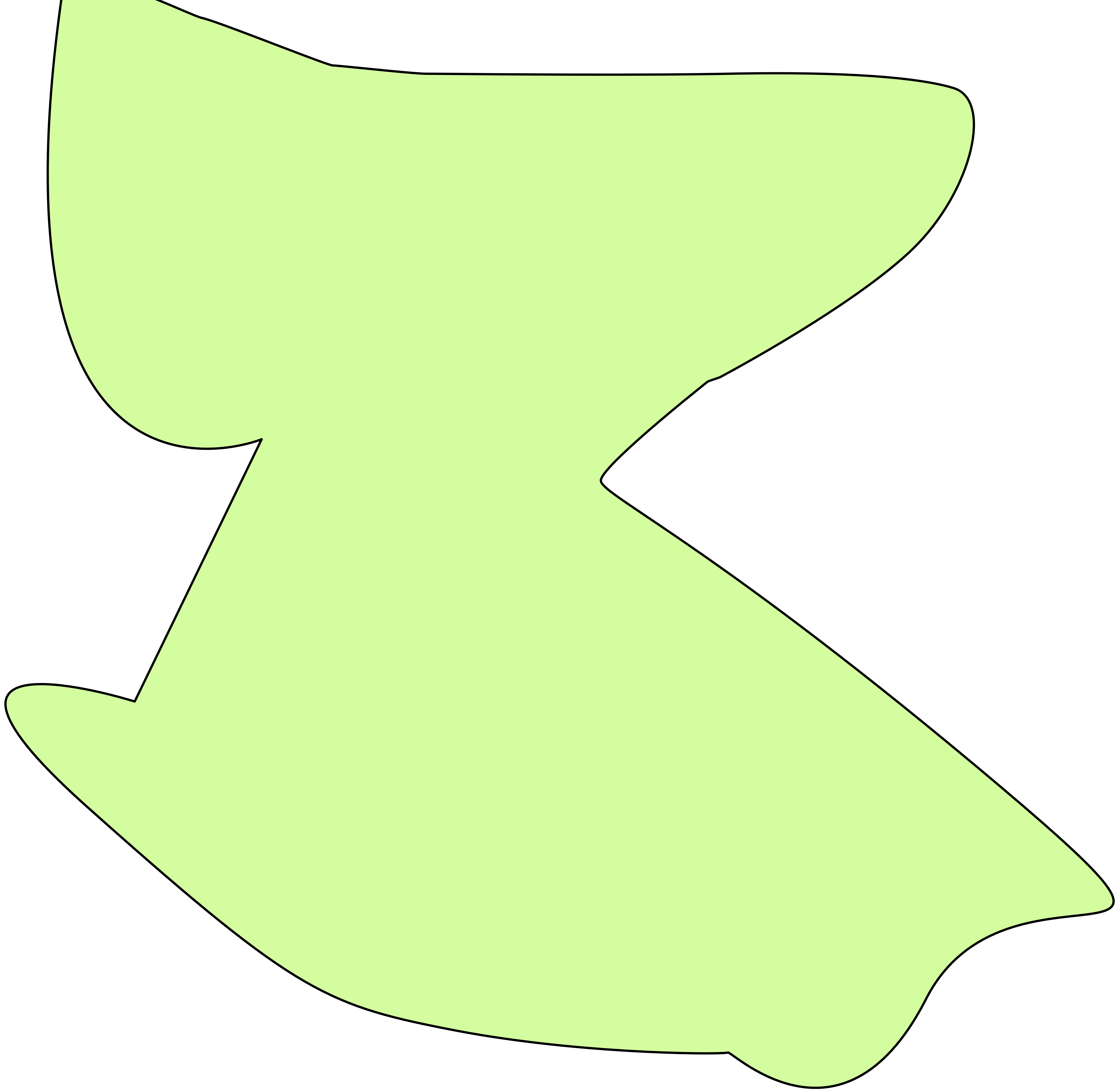
http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg

Examples we will discuss

Merge

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$(A[1..n], m)$:

  $i \leftarrow 1$; $j \leftarrow m+1$

  for $k \leftarrow 1$ to n

    if $j > n$

      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$

    else if $i > m$

      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$

    else if $A[i] < A[j]$

      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$

    else

      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$

  for $k \leftarrow 1$ to n

    $A[k] \leftarrow B[k]$

jeff erickson

5	2	4	7	1	3	2	6

```
merge-sort (A, p, r)
  if p < r
    q ← ⌊(p + r)/2⌋
    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

5	2	4	7	1	3	2	6

5	2	4	7

1	3	2	6

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$(A[1..n], m)$:

$i \leftarrow 1$; $j \leftarrow m+1$

for $k \leftarrow 1$ to n

  if $j > n$

    $B[k] \leftarrow A[i]$; $i \leftarrow i+1$

  else if $i > m$

    $B[k] \leftarrow A[j]$; $j \leftarrow j+1$

  else if $A[i] < A[j]$

    $B[k] \leftarrow A[i]$; $i \leftarrow i+1$

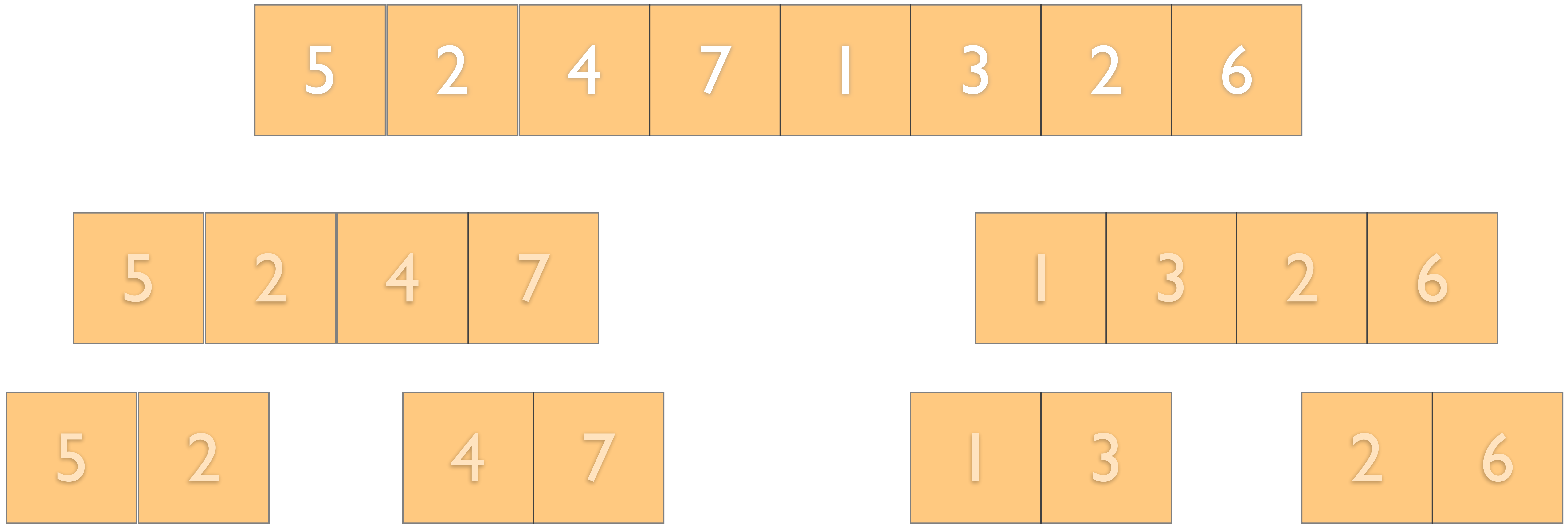
  else

    $B[k] \leftarrow A[j]$; $j \leftarrow j+1$

for $k \leftarrow 1$ to n

  $A[k] \leftarrow B[k]$

jeff erickson

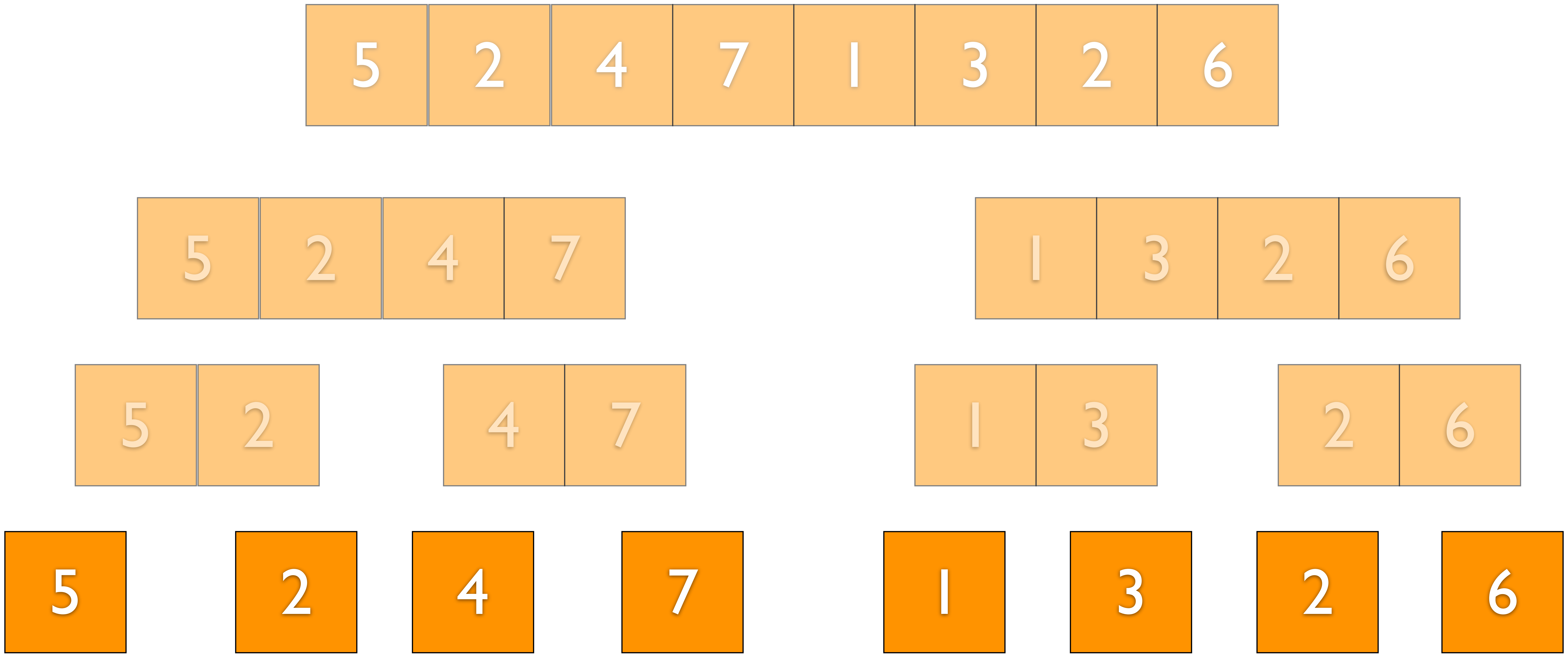


```
merge-sort (A, p, r)
  if p < r
      q ← ⌊(p + r)/2⌋
      merge-sort (A, p, q)
      merge-sort (A, q + 1, r)
      merge(A, p, q, r)
```

```
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



merge-sort (A, p, r)

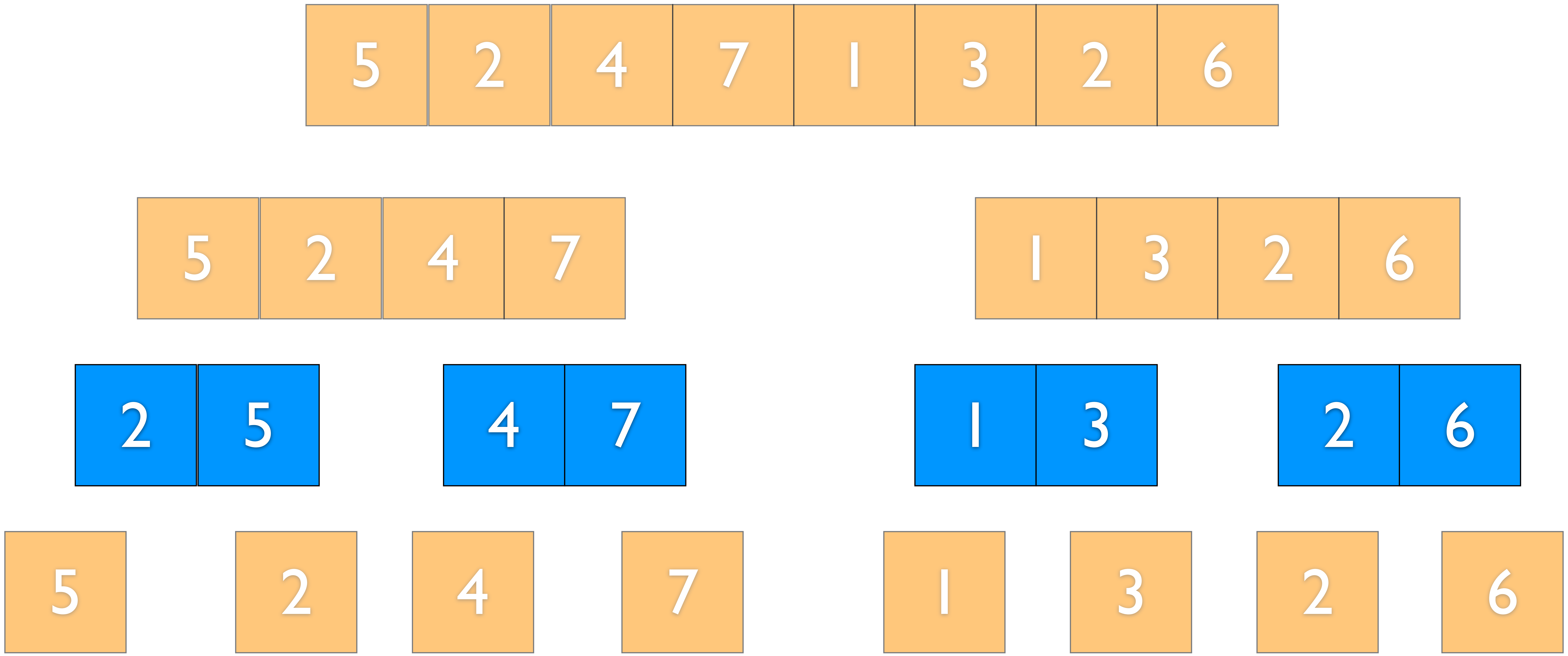
if $p < r$

$q \leftarrow \lfloor (p+r)/2 \rfloor$

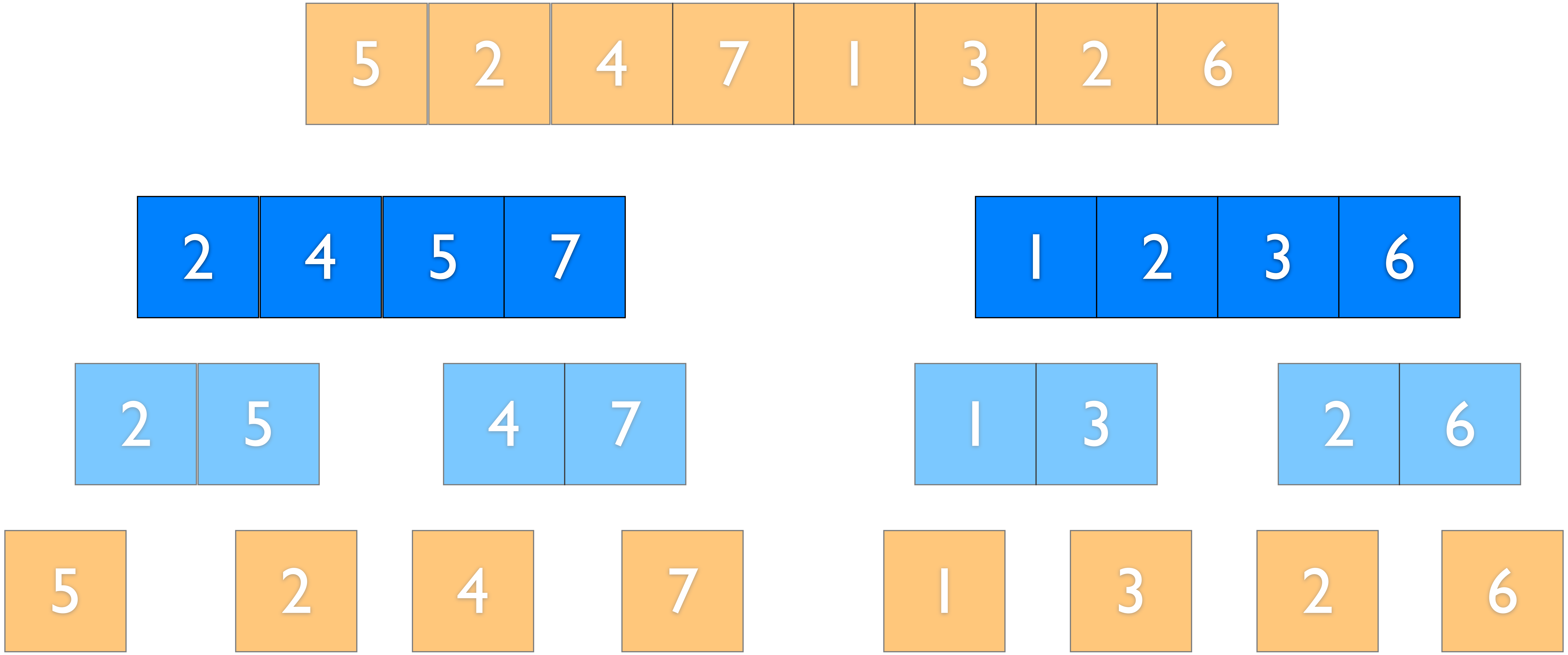
merge-sort (A, p, q)

merge-sort $(A, q+1, r)$

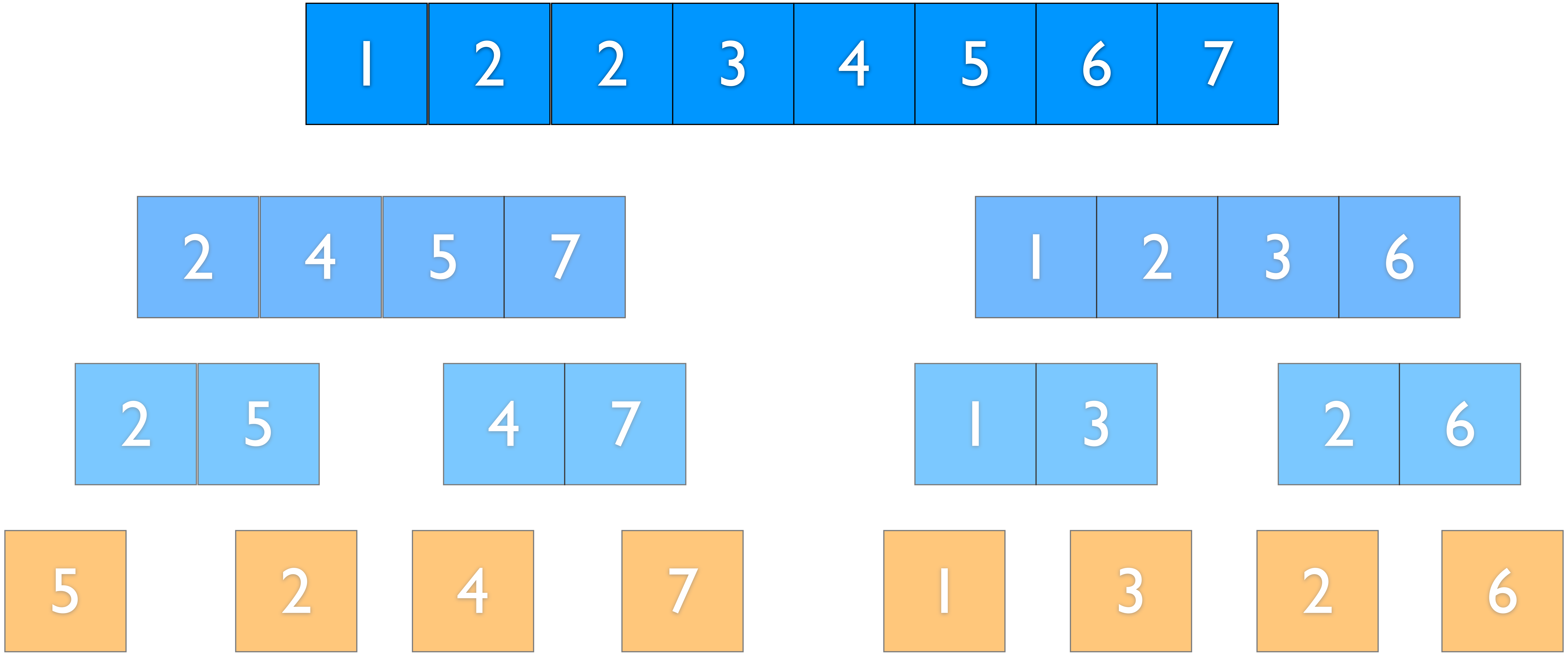
merge(A, p, q, r)



merge-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-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-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)

$$T(n) = 2T(n/2) + O(n)$$
$$= \Theta(n \log n)$$