

GPS 5 Mon Oct 5

HW 5 Tue Oct 6

Exam solutions available. Grades by Tue Oct 13.

Divide-and-conquer:

- divide into smaller instances of the same problem
- delegate smaller instances to the Recursion Fairy
- combine the solutions to the subproblems.

Quicksort (Hoare '61):

Input:	S	O	R	T	I	N	G	E	X	A	M	P	L
Choose a pivot:	S	O	R	T	I	N	G	E	X	A	M	P	L
Partition:	A	G	O	E	I	N	L	M	P	T	X	S	R
Recurse Left:	A	E	G	I	L	M	N	O	P	T	X	S	R
Recurse Right:	A	E	G	I	L	M	N	O	P	R	S	T	X

QuickSort($A[1 \dots n]$):

if $n > 1$

 Choose a pivot element $A[p]$

$r \leftarrow \text{Partition}(A, p)$

 QuickSort($A[1 \dots r - 1]$)

 QuickSort($A[r + 1 \dots n]$)

rearrange A so its

$< \text{pivot}$ | pivot | $> \text{pivot}$

returns r : the final index
of the pivot
(its rank)

Partition($A[1 \dots n], p$):

 swap $A[p] \leftrightarrow A[n]$

$\ell \leftarrow 0$

 for $i \leftarrow 1$ to $n - 1$

 if $A[i] < A[n]$

$\ell \leftarrow \ell + 1$

 swap $A[\ell] \leftrightarrow A[i]$

 swap $A[n] \leftrightarrow A[\ell + 1]$

 return $\ell + 1$

Run time could be as bad as

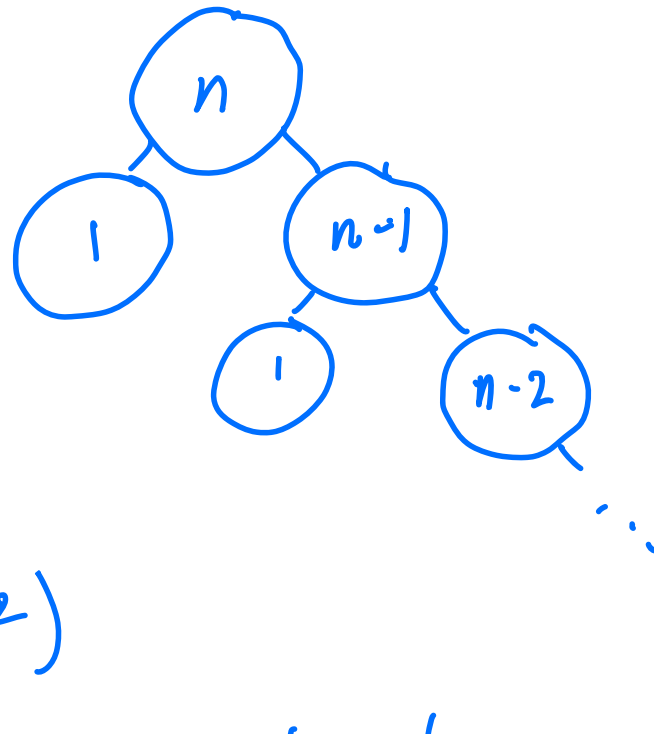
$$T(n) = O(n) + T(n-1) + T(0)$$

if $r = 1$

↑ rank of pivot



$n-1$ elements!



could be a "path"
of depth n

Sums to $O(n^2)$

can happen in practice!

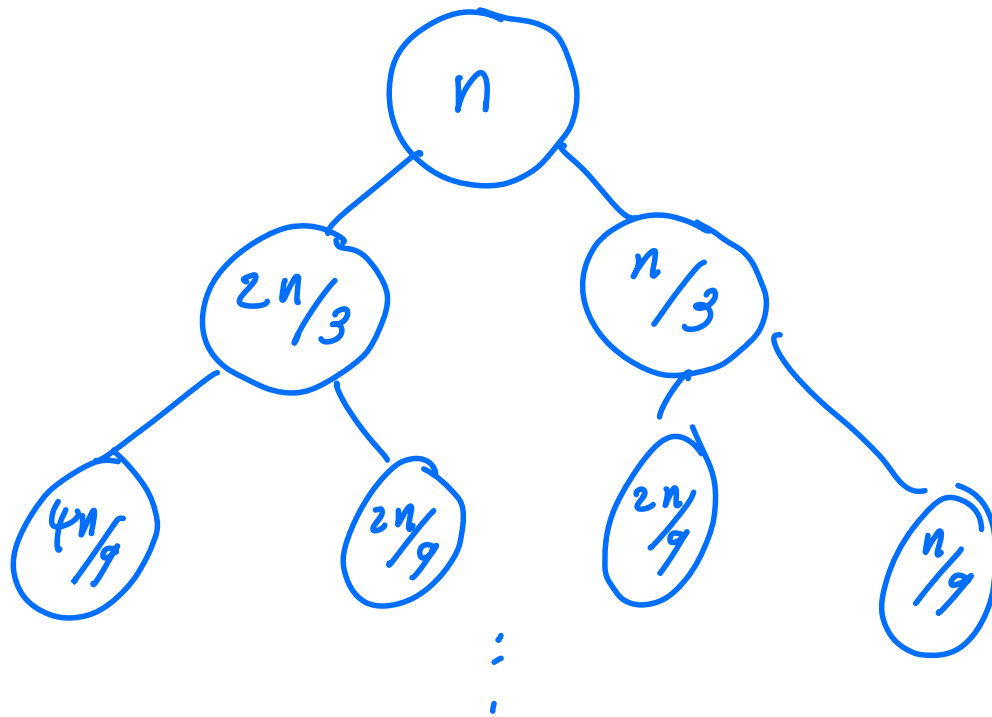
Worst-case analysis

Typically r is near $n/2$, $> A[p]$

Suppose $n/3 \leq r \leq 2n/3$.



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



n

n

n

Each level sums to n ...
At most $\log_{3/2} n$ levels

$$T(n) \leq O(n \log_{3/2} n) = O(n \log n)$$

Assuming $n/3 \leq r \leq 2n/3$!!!

times can multiply n by $2/3$ before reaching 1
 $\equiv$ # times can divide by $3/2$.

Can we find such a pivot?

The median? : element of rank $\lceil n/2 \rceil$

Could do mergesort & return $A[\lceil n/2 \rceil]$

$O(n \log n)$ time

Getter?

median? (rank $k = \lceil n/2 \rceil$)

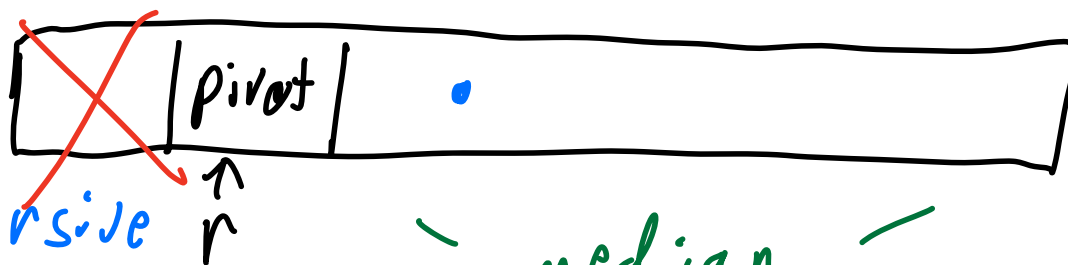
$A[1..n]$:



What if we pick a pivot &

Partition(A, p)?

$r \leftarrow \text{Partition}(A, p)$



Is there a good recursive
call?

median

(assuming $r = \lceil n/2 \rceil$)

Recurse on the half holding the
median!

But rank of median changes in right half.

Selection: Given $A[1..n]$ & a value k

$1 \leq k \leq n$.

Find the element of rank k .

```
QuickSelect( $A[1 .. n]$ ,  $k$ ):
```

```
  if  $n = 1$ 
```

```
    return  $A[1]$ 
```

```
  else
```

```
    Choose a pivot element  $A[p]$ 
```

```
     $r \leftarrow \text{Partition}(A, p)$ 
```

```
    if  $k < r$ 
```

```
      return QuickSelect( $A[1 .. r - 1]$ ,  $k$ )
```

```
    else if  $k > r$ 
```

```
    return QuickSelect( $A[r + 1 .. n]$ ,  $k - r$ )
```

```
    else
```

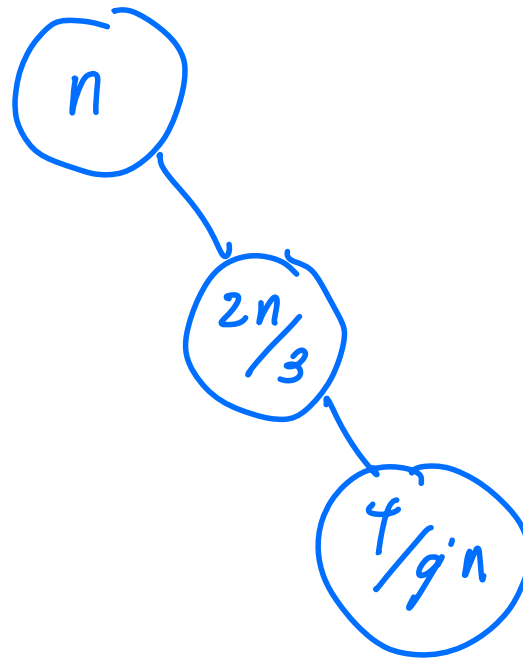
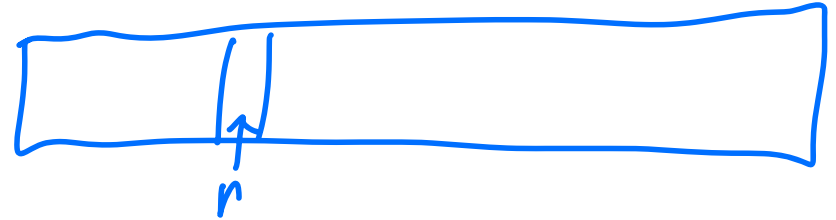
```
      return  $A[r]$ 
```

subarray $A[r+1..n]$
missing smallest
 r elements

Time $T(n) = O(n) + T(n-1)$ in worst case!
 $= O(n^2)$

If $n/3 \leq r \leq 2n/3$

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



n

$2n/3$

$4n/9$

decreasing geometric proportional to largest term

$$T(n) = O(n)$$

Median of Medians Selection BFPRT '70.

```
MomSelect( $A[1 \dots n], k$ ):  
  if  $n \leq 25$  (or whatever)  
    use brute force  
  else  
     $m \leftarrow \lceil n/5 \rceil$   
    for  $i \leftarrow 1$  to  $m$   
       $M[i] \leftarrow \text{MedianOfFive}(A[5i - 4 \dots 5i])$   
     $mom \leftarrow \text{MomSelect}(M[1 \dots m], \lceil m/2 \rceil)$   
     $p \leftarrow$  the index of  $mom$  in  $A$   
     $r \leftarrow \text{Partition}(A, p)$   
    if  $k < r$   
      return  $\text{QuickSelect}(A[1 \dots r - 1], k)$   
    else if  $k > r$   
       $\text{QuickSelect}(A[r + 1 \dots n], k - r)$   
    else  
      return  $A[r]$ 
```

← divide A into blocks of 5, find their medians & recursively get median of medians

“approximate median Fairy”

$$T(n) = O(n) + T(n/5) + ?$$

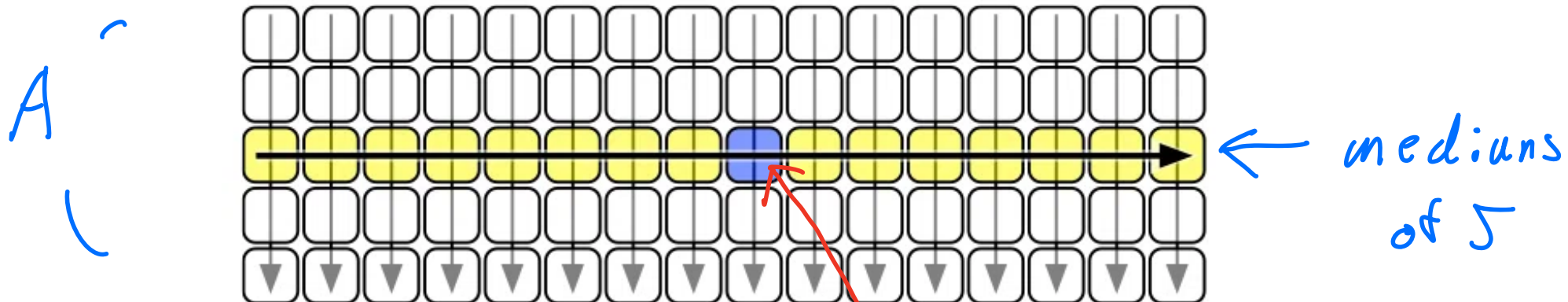
Imagine...

rearrange A into a $5 \times n/5$ grid

each column is one of the 6 blocks

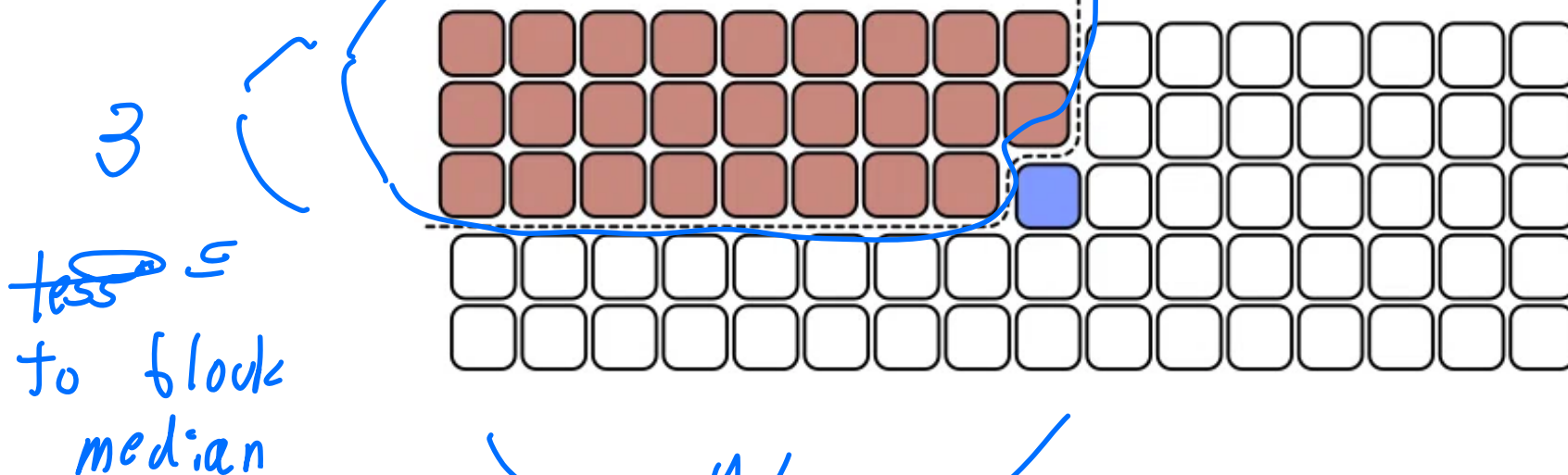
sort each column from ^{small}
 $\downarrow$
 big

& then sort the columns by their medians



all < mom

median of medians
(mom)

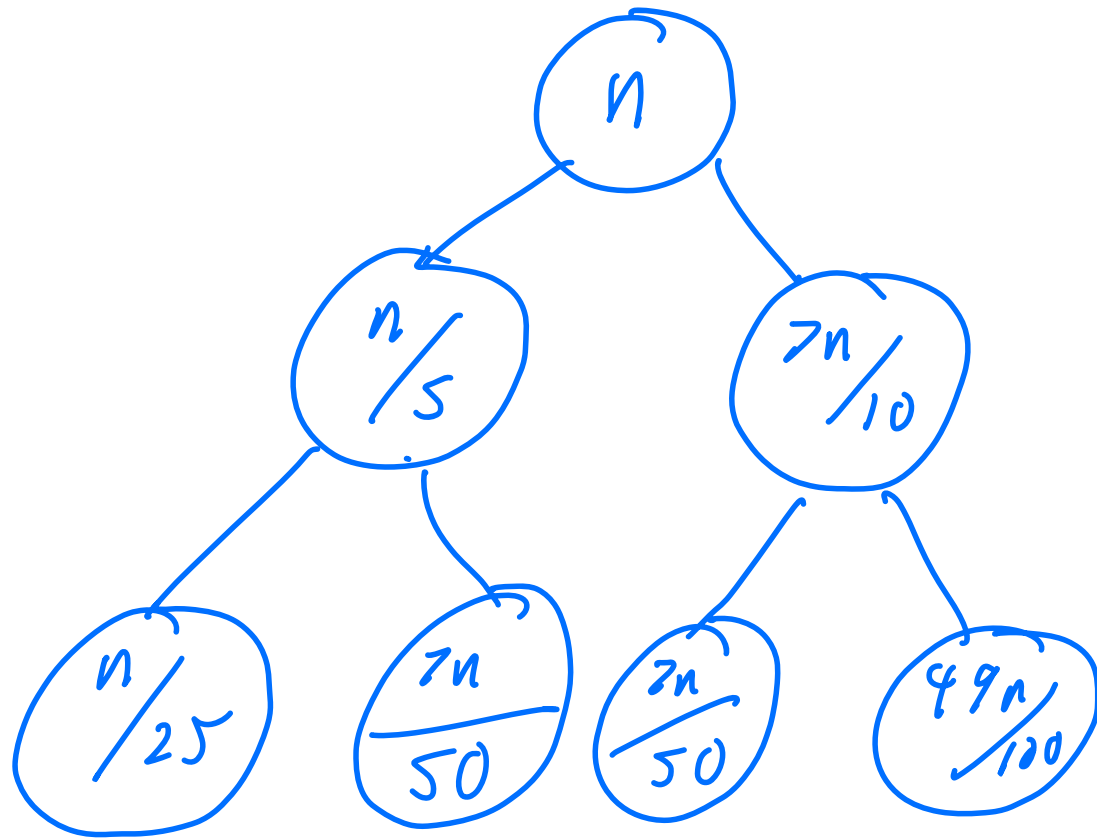


so $\geq \frac{n}{10} \cdot 3 = \frac{3n}{10}$ elements are less than mom

$\Rightarrow \frac{n}{10}$ elements greater than mom

So second call has size $\leq 7^n/10$

$$T(n) \leq O(n) + T(n/5) + T(7^n/10)$$



same n

$$\left(\frac{9}{10}\right) \cdot n$$

$$\left(\frac{81}{100}\right) \cdot n$$

level i is $\left(\frac{9}{10}\right)^i \cdot n$

$$\text{so } T(n) = O(n)$$

For lab tomorrow: You can multiply
two n -digit numbers
using $O(n^{\log_2 3}) = O(n^{1.59})$
single-digit multiplications.