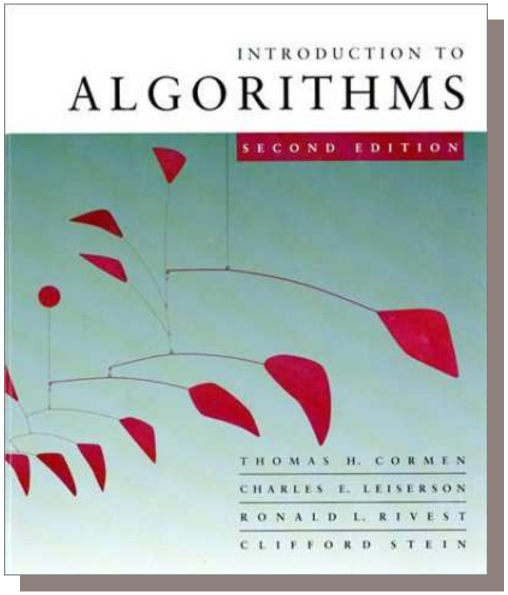
Data Structures and Algorithms
CSE 465



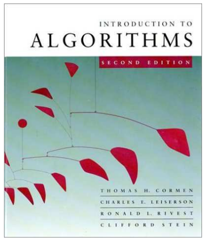
LECTURE 10

- **Order Statistics**
- **Randomized Selection**

Sofya Raskhodnikova and Adam Smith

2/7/2007

S. Raskhodnikova and A. Smith. Based on notes by E. Demaine and C. Leiserson



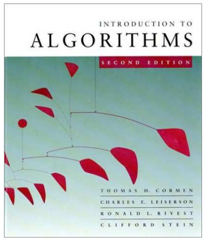
Review questions

You are given an array of integers from 0 to 9 where each integer appears the same number of times.

- You pick an array element at random (with replacement). What is the expected number of elements you have to pick in order to see a 0? (Answer: 10, expectation of a geometric random variable with p=1/10)
- Let I be an indicator random variable defined as

$$I = \begin{cases} 1 & \text{if in the first 3 trials you pick 4,6,5, respectively;} \\ 0 & \text{otherwise.} \end{cases}$$

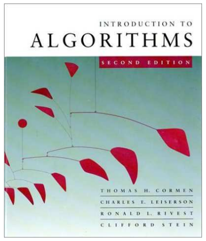
Find $E[I]$. **(Answer: 1/1000, for an indicator random variable E[I]=Pr[I=1])**



Background: Expectation

The expectation of a random variable is the first thing you want to find out about a random variable when you are trying to understand its behavior. Roughly, the expectation is the average, where each value is weighted according to the probability with which it comes up:

- $E[A] = \sum_{i \in \text{range of } A} \left(\text{Prob}[\, A=i \,] \cdot i\right)$



Order statistics

Select the ith smallest of n elements (the element with ***rank*** i).

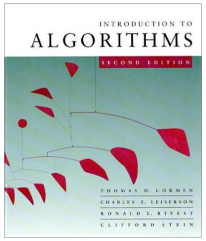
- $i = 1$: ***minimum***;
- $i = n$: ***maximum***;
- $i = \lfloor (n+1)/2 \rfloor$ or $\lceil (n+1)/2 \rceil$: ***median***;
- $i = \lfloor 0.9n \rfloor$: ***90th percentile***.

Naive algorithm: Sort and index ith element.

Worst-case running time $= \Theta(n \lg n) + \Theta(1)$

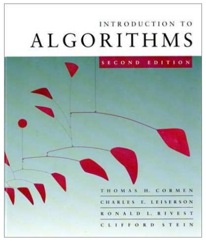
$= \Theta(n \lg n)$,

using merge sort (*not* quicksort…).



How fast can we find order statistics?

- Can find minimum and maximum in $\Theta(n)$ time (easy)
- Today we will learn how to find other statistics, e.g., median, in expected linear time.

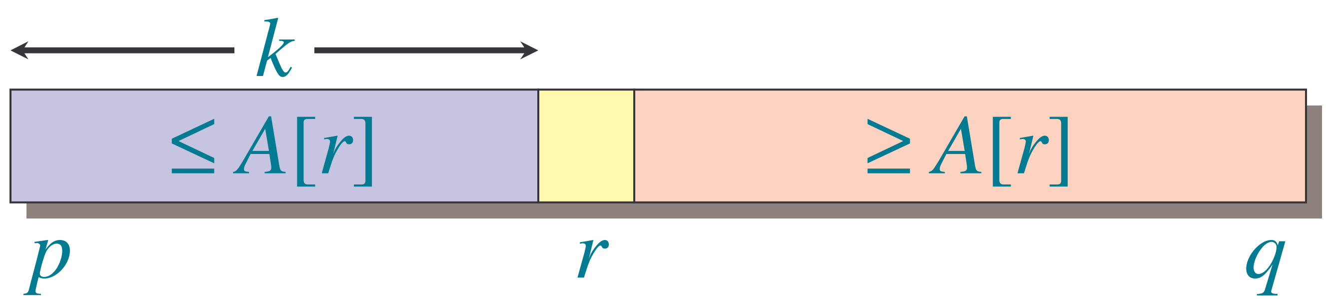


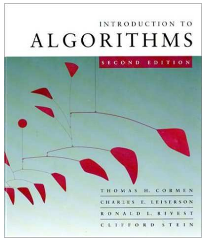
Main idea for the algorithm

- Partition the array *A* using a random element as a pivot (like in QuickSort)
- Decide in which part the *i*-th smallest element resides
- Recurse on that part

Randomized divide-and-conquer algorithm

RAND-SELECT(A, p, q, i) ▹ ith smallest of $A[p\,..\,q]$
  if $p = q$ **then return** $A[p]$
  $r \leftarrow$ RAND-PARTITION(A, p, q)
  $k \leftarrow r - p + 1$ ▹ $k = \text{rank}(A[r])$
  if $i = k$ **then return** $A[r]$
  if $i < k$
    then return RAND-SELECT$(A, p, r - 1, i)$
    else return RAND-SELECT$(A, r + 1, q, i - k)$





Recall: Randomized partition

RAND-PARTITION(A, p, q)

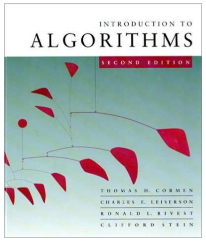
  ▷ Partition $A[p \mathinner{.\,.} q]$ around random pivot

$t \leftarrow$ RANDOM(p,r)

  ▷ t is a random integer between p and r

Exchange $A[p] \leftrightarrow A[t]$

return PARTITION(A, p, r)



Example

Select the $i = 7$th smallest:

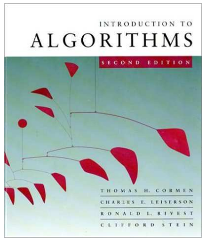
6	10	13	5	8	3	2	11	$i = 7$

pivot

Partition:

2	5	3	6	8	13	10	11	$k = 4$

Select the $7 - 4 = 3$rd smallest recursively.



Intuition for analysis

(Assume that all elements are distinct.)

Lucky: each pivot partitions the array into parts of size at most 3/4 of the original.

$$T(n) = T(3n/4) + \Theta(n)$$
$$= \Theta(n)$$

$$n^{\log_{4/3} 1} = n^0 = 1$$

Master Theorem, CASE 3

Worst case: always partition around min or max

$$T(n) = T(n-1) + \Theta(n)$$
$$= \Theta(n^2)$$

arithmetic series

Worse than sorting!