6.006
Rivest
L26.1
12/9/08

Admin: Final exam Mon 12/15 1:30-4:30
in Johnson
(See TA's if you have conflict!)

Relevant Readings: CLRS Chapters 5, 7, 34, 35

(This material will not be on final.)

Outline: "On Beyond 6.006"

- Courses
- Topics

6.006
Rivest
L26.2
12/9/08

Courses

6.046 "Intermediate Algorithms" (Leiserson/Goemans)
more advanced algs. & analysis techniques
less coding
randomized algs, data structures

6.047 Computational Biology (Kellis)
more string-based algorithms for genetic data
phylogeny

6.854 Advanced Algorithms (Karger)
intense survey of field

6.850 Geometric Computing (Indyk)
points/lines/polygons/meshes ...

6.851 Advanced Data Structures (Demaine)
e.g. sublogarithmic performance

6.852 Distributed Algorithms (Lynch)
reaching consensus in a network with faults

6.855 Network Optimization (Schulz)
optimization in a graph — beyond shortest paths

6.006
Rivest
L26.3
12/9/08

6.856: Randomized Algorithms (Karger)
how randomness can make algs simpler & faster

6.857: Network & Computer Security (Rivest)
(applied cryptography)

6.885: Geometric Folding Algorithms
how to fold one shape into another

Other theory classes:

6.045: Automata, Computability, & complexity →
6.840: Theory of Computing (Sipser)
6.841: Advanced Complexity Theory (Sudan) → merging?
6.842: Randomness & Computation (Rubinfeld)?
6.080: Great Ideas in Theoretical Comp Sci. → (Aaronson)

Topics

6.006
Rivest
L26.4
12/9/08

① Randomness

flipping coins could help!?

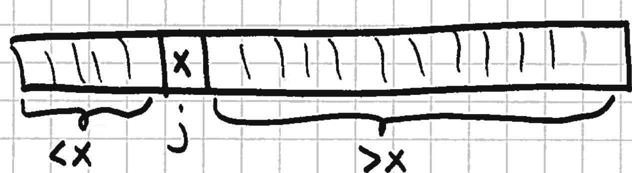
randomness in algorithm, not same as
randomness in input

aim for good expected running time

Example: Quicksort

given $A[1..n]$ to sort

- Pick a random i, $1 \le i \le n$
- let $x = A[i]$ (= "pivot")
- partition A so that (in time $\Theta(n)$)



(e.g. if all element distinct)

- sort $A[1..j-1]$ and $A[j+1..n]$ recursively

sizes of subproblems are random variables

but: this works extremely well!!

good $\Theta(n \lg n)$ expected running time.

Analysis in 6.046

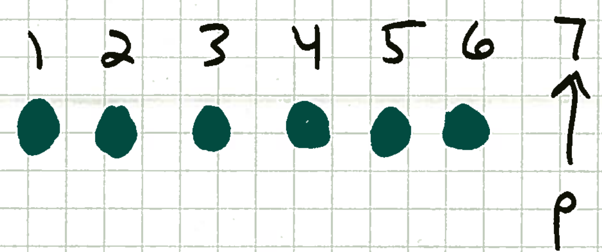
(Randomness con'd...)

6.006
Rivest
L26.5
12/9/08

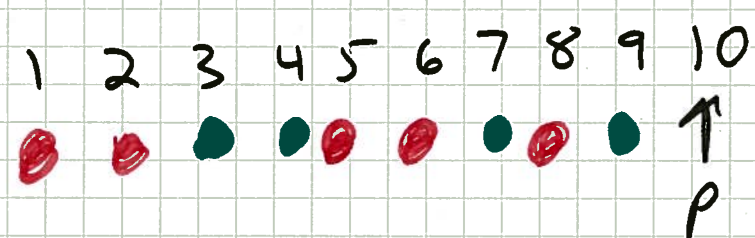
Example 2: Primality testing

When testing a number p for primality, there is a "color test" (details omitted) for integers a, $1 \le a < p$ such that

- if p is prime, then all such a's are green:



- but if p is not prime, at least 1/2 the a's are red



Trouble is: pattern looks pretty "random", no easy way to find a red a deterministically.

So: try 50 a's at random

all green ⇒ say "prime" (we hope)

any reds ⇒ say "not prime" (for sure!)

error controllable by adjusting the "50"

6.006
Rivest
L26.6
12/9/08

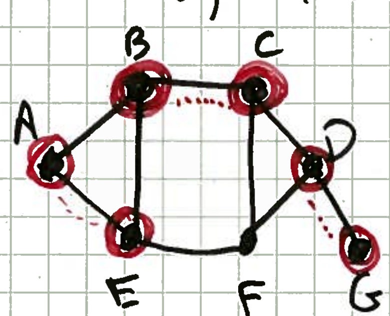
Approximate algorithms

Once we open the door to algorithms that are not always right, we can do other things, e.g. "just get close" in other ways.

Consider again Vertex Cover: Given an undirected graph $G=(V,E)$, we want to find a small set of vertices to color red so that each edge touches at least one red vertex.] NP-hard, NP-complete

Consider following algorithm:

- Pick an edge arbitrarily (or at random)
- color both endpoints red
- remove from consideration both endpoints & all edges that touch them (they're covered now)
- repeat until no edges left



BC
AE
DG

Claim:

Vertex cover found is at most twice optimal size

Proof:

Each edge picked must have at least one endpoint in minimum cover. (edges picked don't touch)

Finding minimum cover is hard; what is best approximate cover? (can we beat factor of 2)?

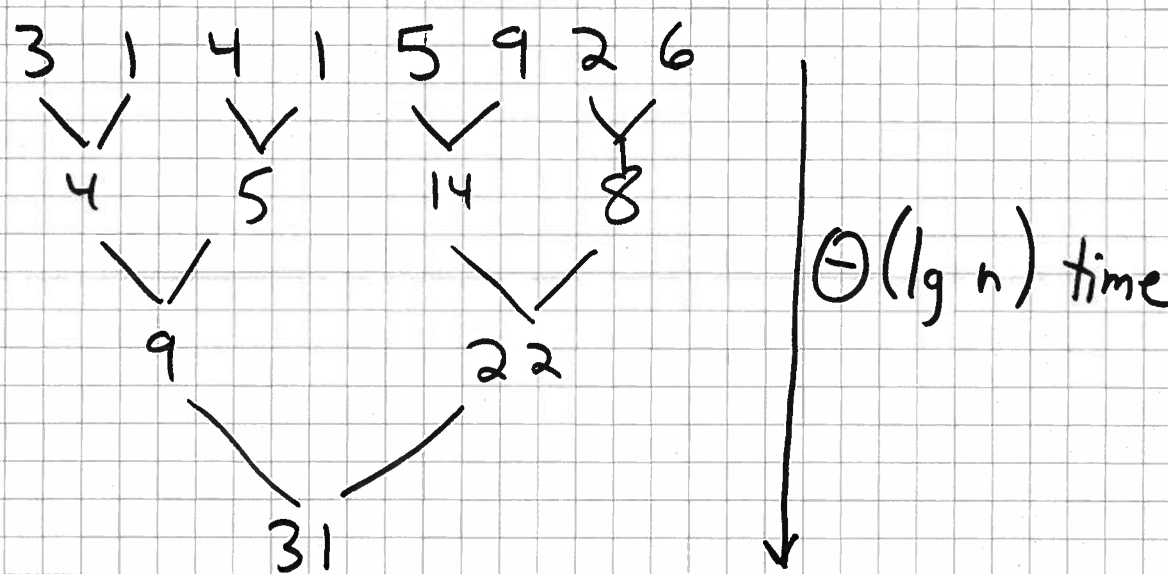
[seems very hard to do so!]

Whole field of approximation algorithms!!

6.006
Rivest
L26.7
12/9/06

Parallel Computing

① On an infinitely parallel computer, how long to add up n numbers?



② How long to computer dot-product of two length n vectors $x \cdot y = \sum_{i=1}^{n} x_i \cdot y_i$

Ans: $\Theta(\lg n)$ time (1 to mpy, $\lg n$ to add)

Parallel Computing (cont.)

6.006
Rivest
L26.8
12/9/08

(3) How long to compute product of two $n \times n$ matrices?

$$A \cdot B = C$$

Ans: $\Theta(\lg n)$ time
(do n^2 dot-products in parallel...)

(4) How long to compute shortest paths in an n-vertex graph?

Ans: $\Theta(\lg^2 n)$ time

Do $\lg n$ "matrix multiplies" $A, A^2, A^4, \ldots, A^n$
where inner product $x \cdot y$ means $\min_i (x_i + y_i)$
starting from given edge-weight matrix (0's on diagonal) A.

• • • See you at the final!