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

$0 \le f(n) \le c\,g(n)$

$f(n) = o(g(n))$ upper

$0 \le f(n) < c\,g(n)$

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

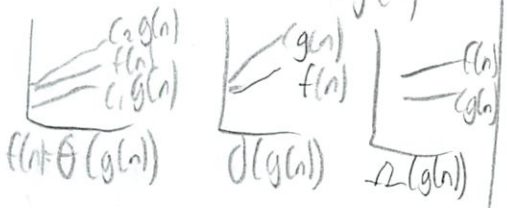
$0 \le c\,g(n) \le f(n)$

$f(n) = \omega(g(n))$ lower

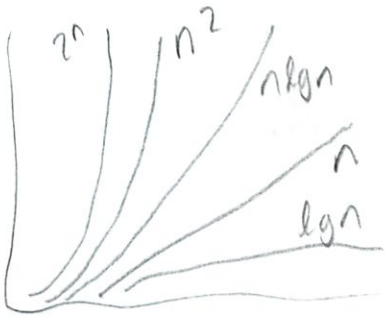
$0 \le c\,g(n) < f(n)$

$f(n) = \Theta(g(n))$ asy tight = upper + lower

$0 \le c_1 g(n) \le f(n) \le c_2 g(n)$



	upper	both	lower
$=$	$\rightarrow$	Θ	$\leftarrow$
$\le$	O	$\wedge$	Ω
$<$	o	$\times$	ω



If correct, always pass
If incorrect, $P[\text{fail}] \ge \frac{1}{2}$
$AB \ne C$ $\quad P_{error} = \frac{1}{2^k}$

$B \cdot r$
$A \cdot (B \cdot r)$
$C \cdot r$
$A \cdot B \cdot r \stackrel{?}{=} C \cdot r$

6.046 Exam 1

Interval Scheduling $O(n \lg n)$

Greedy

weighted: DP

$\max(w_i + opt(p^f(i)))$

Diff servers: impossible

Monte Carlo small prob incorrect

deterministic runtime

Las Vegas always correct

random makes faster

Convex Hull all points on or inside

Graham $O(n \lg n)$

Jarvis: $O(nh)$ - h = vertices on hull

incremental $O(n \lg n)$

divide + conquer $O(n \lg n)$

prune + search: $O(n \lg h)$

Median Finding $O(n)$

cols w/ 5 items each - median of that to get x → the center which we can recurse right + left

Quicksort $\Theta(n^2)$

Random QS $\Theta(n \lg n)$

Pareto $|L| \le \frac{3}{4}|A|$

$\Theta(n \lg n)$

Skip Lists $O(\lg n)$ finding

randomized data structure

14 → 50 → 70

14 → 34 → 50 → 66 → 70

14 23 34 42 50 59 66 72 70

EV = weighted avg of all possibilities

Freivalds $A \cdot B = C$ matrix

get random r: from {0,1}

some is bad

$O(n^{2.3727})$

normal multiply $O(n^2)$

FFT " $O(n \lg n)$

10/10/2012

12/17/2012

FFT

Convert from coeff to pt value form

then can add or multiply $O(n)$

but this conversion must be efficient

choose $\pm x_0, \pm x_1, \ldots \pm x_{\frac{n-1}{2}}$

$A(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \ldots$

$A(-x_i) = a_0 - a_1 x_i + a_2 x_i^2 + \ldots$

$A_{even} = a_0 + a_2 x + a_4 x^2 + \ldots$

$A_{odd} = a_1 + a_3 x + a_5 x^2 + \ldots$

$A(x_i) = A_{even}(x_i^2) + x_i A_{odd}(x_i^2)$

$A(-x_i) = A_{even}(x_i^2) - x_i A_{odd}(x_i^2)$

reduce to 2 subproblems, each deg $< \frac{n}{2}$

Better notation

$e^{iu} = \cos(u) + i \sin(u)$

Sols to $w^n = 1$ are $e^{2\pi i k/n}$ for $k = 0, \ldots n-1$

Alg FFT(A_n)

if $n = 1$, return $A(1)$

write $A(x)$ as

$A(x) = A_{even}(x^2) + A_{odd}(x^2)$

Call FFT($A_{even}, \frac{n}{2}$)

call FFT($A_{odd}, \frac{n}{2}$)

Compute A at n powers of w_n

$A(w_n^j) = A_{even}(w_n^{2j}) + w_j A_{odd}(w_n^{2j})$

return $A(w_n^0) A(w_n^1) \ldots$

$w_n = e^{2\pi i/n}$

$w_{n,j} = -w^{n/2+j} = e^{\frac{2\pi i(n/2+j)}{n}}$

$= (e^{2\pi i/2})(e^{2\pi i j/n})$

$= -w_n^{n/2+j}$

$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & & \\ & & & & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Value rep

Vandermonde matrix M

Coefficient rep

Eval multiply by M

interp multiply by M^{-1}

Only at pts of unity

Polynomial identities

deterministic $O(n)$
random $O(1)$

Given inputs that are distinct for each possible y, exactly one degree $\le n$ polynomial

$f(x_0) = y_0$

$f(x_1) = y_1$

$\vdots$

$f(x_n) = y_n$

eg 2 pts determine a line
3 pts quadratic polynomial

So randomly try evaluating till disprove!

if wrong, may get true + false

DP

1. Characterize optimal sol'n
2. Recursively define the value of an optimal sol from optimal subproblems
3. Compute value of optimal in bottom up fashion

top down: recurse + memoize
bottom up: iterative

What is subproblem?
Memoize!

Master Theorem

$T(n) = aT(\frac{n}{b}) + f(n)$

$\frac{n}{b}$ = # of recursive calls

Look at $n^{\log_b a}$ vs $f(n)$

For each $n^c \cdot (\lg n)^d$

1. Compare c's
   - bigger c
2. Compare d's
   - bigger one $\cdot \lg n$

$n^{\log_b a} \cdot \lg n$

Single Source Shortest Path

Unweighted BFS $O(n+m)$
- queue

non neg weights Dij $O(m+n \log n)$

general Bellman Ford $O(nm)$

acyclic Topo sort + 1 pass B.F. $O(n+m)$

topo sort DFS, sort by finishing time $O(V+E)$

strongly connected from one point to any other

Min Spanning Tree

A is min spanning tree

greedy find an edge so A still ↑
repeat, return A

MST-Kruskal(G, w) $O(E \lg V)$

A = 0 best edge of any set

for each vertex $v \in G.V$
  Make-Set(v)

Sort the edges G.E in nondec order by weight

for each edge $(u,v) \in G.E$
  if FindSet(u) ≠ FindSet(v)
    $A = A \cup \{(u,v)\}$
    Union(u,v)

return A

MST-Prim(G, w, r) $O(E \lg V)$ $O(E + V \lg V)$

for each $u \in G.V$
  u.key = ∞
  u.π = nil

r.key = 0

Q = G.V

while Q ≠ 0
  u = Extract-Min(Q) ← based on key
  for each $v \in G.adj[u]$
    if $v \in Q$ and $w(u,v) <$ key
      v.π = u
      v.key = w(u,v)

finds min edge

min priority queue

Floyd Warshall(w) $\Theta(n^3)$

n = W.rows

$D^{(0)} = W$

for k = 1 to n ← specific points
  let $D^{(k)} = d_{ij}^{(k)}$ be new n×n matrix
  for i to n
    for j to n
      $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

Johnson(w, weight) $O(V^2 \lg V + VE)$

1. Find h s.t. $w_h(u,v) \ge 0$
   Set $h(v) = \delta(s,v)$ w/ Bellman Ford
2. Reweight all edges via
   $w_h(u,v) = w(u,v) + h(u) - h(v)$
3. Run Dij for all source nodes $u \in V$ using w_h
4. Reweight all edges $w(u,v) \in$
   $w_h(u,v) - h(u) + h(v)$

Network Flow

Residual $c_f(u,v) = c(u,v) - f(u,v) > 0$

Final Exam

Max Flow: Ford Fulkerson $O(E|f^*|)$

initialize flow f to 0

while there exists aug pt p in G_f

augment path $f \to p$

return f (picks random - silly!)

Resid G_f

$$C_f(u,v) = \begin{cases} C(u,v) - f(u,v) \\ f(v,u) \\ 0 \end{cases}$$

Cut $|f|$ always the same

Max flow = min cut

f is max flow if G_f has no aug pths

Edmond Karp $O(VE^2)$

Smart pick shortest path $s \to t$ in G_f w/ BFS

Upward Critical when ↑ capacity, ↑ max flow

U = set nodes in G_f reachable s

V = t

edge in both u, v

Downward critical ↓ max flow

no path u to v in G_f

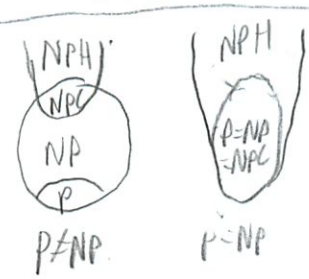
P, NP | P n^k = poly time

1. Decision → Yes/No solvable
2. Search → find an object
3. Optimization → find best

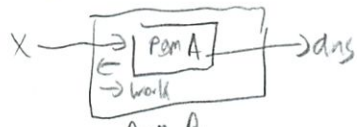
NP nondeterministic poly time verifiable

NP-hard if all problems in NP are poly reducable to it. Can solve any problem in NP

NP-complete if it is in NP and is NP-hard



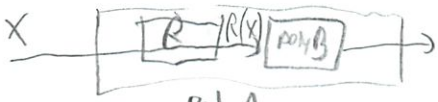
Cook have $A \in P$, want to show $B \in P$



Showed that every NP problem reduces to SAT

Karp know $A \in NPC$, show $B \in NPC$

$A \propto B$ know $\propto$ want to prove



If A is NPH, Then B is NPH

A is impossible, If we can solve A by reducing it to B (black box) then done magic

Since no magic, no B black box, so B is NPH

Given input to A, create an input for B

if A yes → B yes

A no → B no

General Technique know A is NPC, show B is NPC

1. Prove verifier to show $B \in NP$
2. Reduce A to B
   a) give poly time algo R s.t. if x is input for A, then R(x) input for B
   b) Show that if $x \in A$, then $R(x)$ in B
   c) Show that if $R(x)$ in B, Then $x \in A$
3. Conclude B is NPC

Show any NPC problem reduces to it

Bellman Ford for each vertex, for each edge, relax

Dij while queue ≠ ∅, extract min, for each edge, relax

Linear Programming

12/14/12

$\min\ x_1 + x_2 + x_3 + x_4$

s.t. $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$

etc

$x_1, x_2, x_3, x_4 \geq 0$

Std Form

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t. } \sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i = 1 \ldots m$$

$$x_j \geq 0 \text{ for } j = 1 \ldots m$$

$\max\ c^T x$
s.t. $Ax \leq b$
$x \geq 0$

1. Negate coeff in objective
2. Non reg $x_j \to x_j' - x_j''$
3. Variables non neg $-3x \to -3x' + 3x''$
4. Convert equality to 2 inequalities
5. Negate constraints

Totally Unimodular if det of each sq submatrix of A is 0, −1 or 1

Then A will have integral optimum

Slack form Minimize $c^T x$
s.t. $Ax = b$
$x_i \geq 0\ \forall x_i \in X$

Simplex alg

Dual	Primal	Dual
	$\max\ c^T x$	$\min\ b^T y$
	s.t. $Ax \leq b$	s.t. $A^T y \geq c$
	$x \geq 0$	$y \geq 0$

Since $b^T y = c^T x$

Slack $a_{i1}x_1 + \ldots + a_{in}x_n + y_i = b_i$ } m new variables

$y_i \geq 0$ } m new constraints

Std $x_1 + x_2 + x_3 \leq 30$

slack $\underbrace{x_4}_{\text{basic var}} = 30 - \underbrace{x_1 - x_2 - 3x_3}_{\text{non-basic variable}}$

Pivot swap x_1, x_6

$x_1 = \frac{36 - x_2 - 2x_3 - x_6}{4}$

$x_4 = 30 - \left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) - x_2 - 3x_3$

x_4 leaving, x_1 = entering

x_2, x_3, x_6 non basic

$x_1\ x_5\ x_4$ basic

Approx Alg

$\max\left(\frac{C}{C_{opt}}, \frac{C_{opt}}{C}\right) \le \rho(n)$

To prove:
1. Correct
2. Poly time
3. Ans within 2 of optimal

Hamiltonian Cycle - visit each vertex once

triangle ineq $C(u,w) \le C(u,v) + C(v,w)$

vertex cover covers every edge at least once

maximal ind set = 1

Amortized Analysis

Aggregate $\sum$ costs

Accounting $\sum \hat{c}_i \ge \sum c_i$

Potential $\sum \hat{c}_i = \sum c_i + \phi(D_n) - \phi(D_0)$

$\hat{c}_i = c_i + \phi(D_i) - \phi(D_{i-1})$

ϕ = bank bal ≥ 0 prepay, ≤ 0 withdraw

Distributed Algorithms

happen concurrently

Hashing

1. Oneway
2. Collision Resistant (Any)
3. " " (Targeted)
4. Pseudo randomness
5. Non-malleability

Digital Sig

$\sigma = \text{sign}(sk_A, M)$

Verify $\rightarrow (M, \sigma, pk_A) = \{T, F\}$

Sealed bid $c(x) = h(r \| x)$

Encryption

$C = e_k(m)$

$m = d_k(c)$

eavesdropper = passive

M in m = active

key back + forth $\xrightarrow{k_a}$ $\xleftarrow{k_a, k_b}$ $\rightarrow k_b$

$k = (g^b)^a \bmod p$

RSA

1. select 2 large primes p, q
2. $n = pq$
3. Pick e rel prime to $\phi(n) = (p-1)(q-1)$
4. Compute d as $e^{-1} \bmod \phi(n)$
5. Publish $P = (e, n)$
6. Keep secret $S = (d, n)$

$P(M) = M^e \bmod n = C$

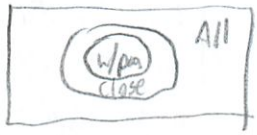
$S(C) = C^d \bmod n = M$

Sub-linear algorithm

Classical
- output is the # that is close to value of optimal sol for given input
- not enough time to build proper sol
- deterministic time alg
- approx ans
- sublinear time

Property testing

output is correct ans for given input or at least other inputs that are close



If correct → must pass

If incorrect → must fail

If ϵ-close can pass $(\frac{2}{3})$ correct or fail (incorrect)

Compression

- lossless
- lossy
- Bloom Filter

$h_1(x_1)\ h_2(x_1)\ h_3(x_1)$

Zero knowledge Proofs

If can tell diff → get right each time

If can't prob get right k times in row is $(\frac{1}{2})^k$

Interactive

Prover - unbounded time

Verifier - poly time

$\max\ 3x_1 + x_2 + 2x_3$

s.t. $x_1 + x_2 + 3x_3 \le 30$

$2x_1 + 2x_2 + 5x_3 \le 24$

$4x_1 + x_2 + 2x_3 \le 36$

$x_1, x_2, x_3 \ge 0$

dual $\min\ 30y_1 + 24y_2 + 36y_3$

s.t. $y_1 + 2y_2 + 4y_3 \ge 3$

$y_1 + 2y_2 + y_3 \ge 1$

$3y_1 + 5y_2 + 2y_3 \ge 2$

$y_1\ y_2\ y_3 \ge 0$

Simplex $z = 3x_1 + x_2 + 2x_3$

$x_4 = 30 - x_1 - x_2 - 3x_3$

$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$

$x_6 = 36 - 4x_1 - x_2 - 2x_3$

$x_4 = 30 - \left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) - x_2 - x_3$

x_1

$z = 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4}$

$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}$

$x_5 = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}$