6.1220 Notes (Triston)

INTRODUCTION

- Greedy Algorithms
- Median Finding Alg

(1) Split into buckets of 5

(2) Pivot at median of the $\frac{n}{5}$ medians

(3) Recurse on either left/right

$T(n)=T(\frac{1}{5}n)+T(\frac{7}{10}n)+O(n)=O(n)$

- Karatsuba's Alg $(O(n^{\log_2 3}))$

$(10^{n/2}A+B)(10^{n/2}C+D)$

$=10^n(AC)+10^{n/2}[(A+B)(C+D)-AC-BD]+BD$

TOOLKIT / PROBABILITY

- Master Thm: $T(n)=aT(\frac{n}{b})+f(n)$

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

~ $f(n)=\Theta(n^{\log_b a}\lg^k n)$: $T(n)=\Theta(n^{\log_b a}\lg^{k+1} n)$

~ $f(n)=\Omega(n^{\log_b a+\epsilon})$ & $af(\frac{n}{b})\le cf(n)$: $T(n)=\Theta(f(n))$

- Chernoff: For $X\sim \mathrm{Bin}(n, \mu/n)$:

$$Pr(X\ge(1+\beta)\mu)\le\left[\frac{e^\beta}{(1+\beta)^{1+\beta}}\right]^\mu\le\begin{cases}e^{-\beta^2\mu/3}, & \beta\le 1\\ e^{-\beta\mu/3}, & \beta\ge 1\end{cases}$$

$$Pr(X\le(1-\beta)\mu)\le\left[\frac{e^\beta}{(1-\beta)^{1-\beta}}\right]^\mu\le e^{-\beta^2\mu/2},\ \beta<1$$

- Union Bound: $Pr(\bigcup E_i)\le\sum Pr(E_i)$
- Markov Ineq: $Pr(X\ge a)\le\frac{\mathbb{E}(X)}{a}$ $(X\ge 0)$
- Chebyshev Ineq: $Pr(|X-\mu|\ge a)\le\frac{Var(X)}{a^2}$

RANDOMIZED

- Quick-Sort Alg

(1) Pick random pivot, divide into L&R.

(2) Quick-Sort L and R. Return (L, p, R).

Runtime: $O(n\log n)$ w/ prob $1-\frac{1}{n}$.

Proof Define 'good' pivot if $\frac{n}{4}\le\text{rank}\le\frac{3}{4}n$.

$Pr(\ge 0.6L$ bad pivots for subproblems including $k)$ $\le e^{-L/150}$ (Chernoff). Pick $L=300\ln(n)$. Bound.

- Binary Matrix Product Check

$AB\ne C\Rightarrow AB\vec{v}\ne C\vec{v}$ w/ prob $\ge\frac{1}{2}$.

Proof Let $AB\vec{n}\ne C\vec{n}$. For any $AB\vec{x}=C\vec{x}$ we have $AB(\vec{n}+\vec{x})\ne C(\vec{n}+\vec{x})$. Injectivity!

AMORTIZED & COMPETITIVE

~ Aggregate (eg. sum over each item)

~ Accounting (give initial 'budget')

~ Potential function **

$\Phi_n\ge\Phi_0(\forall n)$. $\hat{c}_i=c_i+\Delta\phi_i$

- Union-Find

path-compression

union-by-rank/size

~ MakeSet, FindSet, Union

$\Theta(\log n)_{am}\rightarrow\Theta(\alpha(n))_{am}\leftarrow\Theta(\log n)_{am}$

- α-Competitiveness

$cost(A)\le\alpha\cdot cost(OPT)+k$

HASHING / DICTIONARIES

	Space	Ins/Del	Search
Soln. Zero	$O(n)$	$O(1)$	$O(n)$
Dir. Address	$O(\mathcal{U})$	$O(1)$	$O(1)$
Chaining	$O(m+n)$	$O(1)$	$O(1+\alpha)_{avg}$
Ch+Resizing	$O(n)$	$O(1)_{am}$	$O(1)_{avg}$
Open Addr.	$O(m)$	$O(\frac{1}{1-\alpha})_{avg}$	$O(\frac{1}{1-\alpha})_{avg}$
O.A+Resizing	$O(n)$	$O(1)_{am,avg}$	$O(1)_{avg}$
Cuckoo	$O(n)$	$O(1)_{am,avg}$	$O(1)$
Cuckoo+Resizing	$O(n)$	$O(1)_{am,avg}$	$O(1)$

- Uniform Hash Family $\mathcal{H}$:

$\Pr_{h\in_R\mathcal{H}}[h(k)=i]=\frac{1}{m}$ $\forall k\in\mathcal{U}, i\in M$.

- Universal Hash Family $\mathcal{H}$:

$\Pr_{h\in_R\mathcal{H}}[h(k_1)=h(k_2)]\le\frac{1}{m}$ $\forall k_1\ne k_2\in\mathcal{U}$.

- Building Universal Hash Family

m prime. Write all $k\in\mathcal{U}$ as $r=\log_m\mathcal{U}$ digits in base m. Then $\{h_{\vec{a}}(k)=\vec{a}\cdot\vec{k}\mid\vec{a}\in M^r\}$ is universal.

- Open Addressing

$h:\mathcal{U}\times M\rightarrow M$ (Probe seq: perm.)

with uniform (perm) hashing assm.

- Static Dictionary

Want no collisions

Birthday Lemma: If $m\ge n^2$ w/ universal family, $Pr(\text{collision})<0.5$.

↳ 2-Level Hashing: ($O(n)_{avg}$ time)

(1) Hash once. Let n_i=#keys mapped to i. If $\sum n_i^2>4n$, resample. ($Pr<\frac{1}{2}$)

(2) For each $i\in M$, hash to $\{0,\dots,m_i-1\}$ where $m_i=\Theta(n_i^2)$. If collide, resample.

- 2-Way Chaining

→ Two oracles used

→ Put key in the less full bin.

$\mathbb{E}(\text{largest bin size})=O(\lg\lg n)>>O(\frac{\lg n}{\lg\lg n})$

- Cuckoo Hashing

→ Two oracles used

→ Kick existing key to other choice during collision.

→ Cuckoo Graph: $V=M$; $E=$ 2 choices for key.

MINIMUM SPANNING TREES

- MST

Cut Prop: Lightest edge of any "cut" ($\exists!$) is in MST.

Cycle Prop: Heaviest edge of any cycle is not in MST.

Uniqueness: Weights distinct $\Rightarrow$ Unique MST

- Kruskal's Alg

(1) Sort edges by weight

(2) Insert edges if safe starting from lightest (use Union-Find)

Runtime: $O(m\lg m)+O(m\alpha(n))$

- Prim's Alg (~Dijkstra)

→ Grow single tree starting from lightest (use priority Q for unconnected nodes, update connectedness when popping)

Runtime: $O(n\lg n+m)$ with Fibonacci Heap Priority Queue

MAX-FLOW MIN-CUT

- Flow Network

$G=(V,E,s,t,c:E\rightarrow\mathbb{R}_{\ge 0})$

- Residual Network

$G_f=(V,E_f,s,t,c_f)$ where $c_f(e)=c(e)-f(e)$ and $E_f=\{e: c_f(e)>0\}$.

- **Flow Decomposition Thm**

Flow ↗ Flow cycles
Flow ↘ s-t Flow paths.

- **Max Flow–MinCut Thm**

(a) $\exists$cut: $c(S)=f(S)(=|f|)$
(b) f is a maxflow
(c) No s-t paths in G_f

Equiv

- **Ford-Fulkerson Alg** DFS

Keep pushing flow if $\exists$s-t in G_f
$O(mF)=O(mnC)$ (pseudo-poly)

- **Max Bottleneck Path Alg**

Push flow with greatest bottleneck during Ford-Fulkerson
$O(m^2 \lg n \lg(nC))$ (weakly-poly)

- **Edmonds-Karp Alg**

Find s-t in G_f via BFS during FF.
$O(m^2 n)$ (strongly-poly)

LINEAR PROGRAMS

- **Linear Program Duality**

$\max \vec{c}^T\vec{x}$	$\min \vec{b}^T\vec{y}$
$A\vec{x} \le \vec{b}$	$A^T\vec{y} \ge \vec{c}$
$\vec{x} \ge 0$	flip $\vec{y} \ge 0$

- **Strong Duality**

$\vec{c}^T\vec{x} \le \vec{c}^T\vec{x}^* = \vec{b}^T\vec{y}^* \le \vec{b}^T\vec{y}$

- **Complementary Slackness**

$x_i^*((\vec{A}_i)^T\vec{y}^* - c_i) = 0 \quad \forall i$

INTRACTABILITY

- **Verifier for NP** (x=instance, y=certificate)

$V_\pi(x,y)$ $(|y| \le |x|^c)$
→ Runs in $O((|x|+|y|)^{c'})$ time.
→ $\pi(x)=$YES if and only if $\exists |y| \le |x|^{c''}$: $V_\pi(x,y)=$YES.

- **EXP** = {solvable in $2^{poly(n)}$ time}

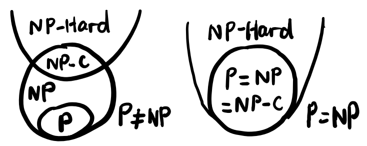
- **$Q \le_p \pi$ (Q reduces to π)**

if there is a poly reduction alg
YES-instances of Q ↦ of π
NO-instances of Q ↦ of π.

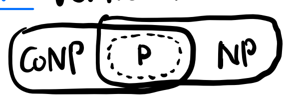
- **NP-Hard**: $Q \le_p \pi \quad \forall Q \in NP$

- **NP-Complete**: NP-Hard ∩ NP.

- 3-SAT, VC, Clique, 3-color, Subset-Sum, Knapsack, 3-NAE-SAT, IP, ... are NP-Complete

- 

- **CoNP**: Verifier for NO-instances



APPROXIMATIONS

- **α-approximation** ($\alpha \ge 1$):

$\frac{OPT(P)}{\mathcal{A}(P)} \le \alpha$ (maximization)
$\frac{\mathcal{A}(P)}{OPT(P)} \le \alpha$ (minimization)

- **Approx. Scheme**: Alg $S(P,\varepsilon)$ which is $(1+\varepsilon)$-approx for all $\varepsilon > 0$.

MULTIPLICATIVE WEIGHTS

- **Online Alg**

(1) Learner picks distribution P_t
(2) Adversary picks costs C_t
(3) Learner picks action (via P_t)
(4) Learner incurs cost, and learns all C_t. Repeat.

- **Expected loss**

$E(C) = \sum_{t=1}^{T}\sum_{a\in A} P_t(a)C_t(a)$

- **Regret** $\frac{1}{T}(E(C)-E(B))$ bench mark

- **Best actions in hindsight** $= B = \sum_{t=1}^{T} \min_{a\in A} C_t(a)$

- **Best fixed action in hs** $= B = \min_{a\in A} \sum_{t=1}^{T} C_t(a)$

- **Expert Predictions**

$m^{(t)} = C_t = 1$ if mistake else 0.

- **Weighted Majority**

$m \le 2(1+\varepsilon)m_i + \frac{2\ln n}{\varepsilon}$.
No vanishing regret (≥ 1)

- **Multiplicative Weights Update**

$E(M) \le (1+\varepsilon)m_i + \frac{\ln n}{\varepsilon}$.
Vanishing regret!

RANDOM WALKS

~Stochastic process: $X=\{X_t : t\in\mathbb{N}\}$
~Markov process: memoryless
~Markov chain: Graph repr of G_X
~Time-homogenous: Same MC $\forall t\in\mathbb{N}$.
~Transition Matrix: $W_{ij} = Pr(X_1=j \mid X_0=i)$
~Stationary Dist: $\vec{\pi} = \vec{\pi}W$
~Communicating class: SCC of G_X
~Recurrent class: w/o outdeg
~Transient class: w/ outdeg (Any random walk vanishes here)
~Class period: GCD of cycle len.
~Aperiodic: Period = 1.
~Uniqueness of $\vec{\pi}$ ⇔ 1 recurrent.
~Convergence ⇔ All recurrent aperiodic
~Detailed balance: $\pi_x W_{xy} = \pi_y W_{yx}$

- **Metropolis-Hastings**

$W_{xy} = g(y|x)P_{acc}(y,x)$, $W_{xx} = 1 - \sum W_{xy}$
$P_{acc}(y,x) = \min\left\{1, \frac{\bar{\pi}(y)g(x|y)}{\bar{\pi}(x)g(y|x)}\right\}$

FAST FOURIER TRANSFORM

- FFT (Want $W\vec{a}$; $W_{ij} = \omega_n^{ij}$)
  (1) If $n=1$, return $\vec{a}$
  (2) $\omega \leftarrow e^{2\pi i/n}$
  (3) $y_{even} \leftarrow FFT(a_0, a_2, \dots, a_{n-2})$
  $y_{odd} \leftarrow FFT(a_1, a_3, \dots, a_{n-1})$
  (4) Return $\vec{y}$ where for $0 \le j < \frac{n}{2}$,
  $y_j = y_{even,j} + \omega^j y_{odd,j}$
  $y_{j+\frac{n}{2}} = y_{even,j} - \omega^j y_{odd,j}$
  modify

- Inverse FFT: $W^{-1} = \frac{1}{n}\overline{W}$
  Runtimes: $O(n \lg n)$.

- Convolution $c_n = \sum_{i=0}^{n} a_i b_{n-i}$

SUBLINEAR ALGORITHMS

- Methods for Sublinear
  1. Classic Approx: Give α-approx output
  2. Property Testing: YES if true; NO if far from true (w/ high prob)

  General Alg:
  (1) Repeat __ times:
  If __, return NO.
  (2) Return YES.

- ε-closeness of G
  Adding $< \varepsilon n \Delta$ edges can make G connected.

- ε-closeness of L
  Deleting $\le \varepsilon n$ items gives a sorted sublist

SKETCHING

- Streaming Alg: Input is seq only passed once/few times.
- Sketch: $C(X)$ = compressed input X.

- Given alg A with time T and space S giving correct expected answer $\mathbb{E}(\hat{\theta}) = \theta$ w/ variance σ^2,
  independent of θ
  $Pr((1+\varepsilon)\text{-approx}) \ge 1 - \frac{\sigma^2/\theta^2}{\varepsilon^2}$

- Mean of Estimates
  $A^m(\varepsilon, \delta)$:
  (1) Repeat A $R = \lceil \frac{\sigma^2/\theta^2}{\varepsilon^2 \delta} \rceil$ times in parallel.
  (2) Output $\hat{\theta} = \frac{1}{R}(\hat{\theta}_1 + \dots + \hat{\theta}_R)$
  $Pr((1+\varepsilon)\text{-approx}) \ge 1 - \delta$
  Time: $O(T)$.
  Space: $O(\frac{\sigma^2/\theta^2}{\varepsilon^2 \delta} S)$

- Median of Means
  $A^{mom}(\varepsilon, \delta)$:
  (1) Repeat $A^m(\varepsilon, \frac{1}{3})$ $R = \lceil 48 \ln(\frac{1}{\delta}) \rceil$ times in parallel.
  (2) Output median.
  $Pr((1+\varepsilon)\text{-approx}) \ge 1 - \delta$
  Time: $O(T + \ln(\frac{1}{\delta}))$
  Space: $O(\frac{\sigma^2/\theta^2}{\varepsilon^2} \ln(\frac{1}{\delta}) S)$.

- t-Wise Independence
  For all distinct $k_1, \dots, k_t \in \mathcal{U}$ and not necessarily distinct $i_1, \dots, i_t \in M$,
  $$\Pr_{h \in_R \mathcal{H}}\left[\bigwedge_{j=1}^{t} h(k_j) = i_j\right] = \frac{1}{|M|^t}$$

- Morris' Alg ($X = 1^n$, $f(X) = n$)
  (1) $C(X) \leftarrow 0$.
  (2) For every 1, increment $C(X)$ w/ prob $2^{-C(X)}$.
  (3) Output $\hat{f}(X) = 2^{C(X)} - 1$.
  Space: $O(\frac{1}{\varepsilon^2} \lg(\frac{1}{\delta}) \lg\lg(\frac{n}{\varepsilon\delta}))$

- FM85 Alg ($f(X)$ = # distinct el.)
  (1) $C(X) \leftarrow 1$.
  (2) $C(X) = \min\{C(X), h(x_i)\}$ $\forall i$ — 2-wise
  (3) Output $\hat{f}(X) = \frac{1}{C(X)} - 1$.
  $\mathbb{E}(\hat{\theta}) = \frac{1}{d+1}$
  $\sigma^2 = \frac{d}{(d+1)^2 (d+2)}$

- KMV Alg
  (1) Pick 2-wise $h: \mathcal{U} \to M$ where $|M| = u^3$.
  (2) $C(X) \leftarrow \emptyset$.
  $k \leftarrow \lceil 24/\varepsilon^2 \rceil$.
  (3) $C(X) = \min_k \{C(X) \cup \{h(x_i)\}\}$ $\forall i$.
  (4) Output
  $$\begin{cases} |C(X)| & \text{if } |C(X)| < k \\ ku^3/\max C(X) & \text{o.w.} \end{cases}$$
  $Pr((1+\varepsilon)\text{-approx}) \ge 2/3$.

- Jaccard Similarity
  $J(X,Y) = |X \cap Y| / |X \cup Y|$.

- Signature of A under h
  $\sigma_h(A) = \min_{a \in A}\{h(a)\}$
  $\Rightarrow Pr[\sigma_h(A) = \sigma_h(B)] = J(A,B)$.

- Similarity/Near neighbor Search
  Given $A_1, \dots, A_n, s, s'$. When A is passed in, output
  → YES if $\exists J(A, A_i) \ge s$
  → NO if $\exists J(A, A_i) < s'$.

- Locally Sensitive Hashing Alg
  → To reduce false negatives:
  (1) Build L signatures to build L perfect hash tables.
  $\therefore Pr(A_i, A_j \text{ same bucket})$
  $\ge 1 - (1 - J(A_i, A_j))^L \xrightarrow{L \to \infty} 1$.
  → To reduce false positives:
  (1) Use t min-hashes instead, i.e. $h_{11}, \dots, h_{Lt}$ with
  $\sigma_{h_\ell}(A) = (\sigma_{h_{\ell 1}}(A), \dots, \sigma_{h_{\ell t}}(A))$
  $\therefore Pr(A_i, A_j \text{ same bucket})$
  $\ge 1 - (1 - J(A_i, A_j)^t)^L$
  Tweak t and L for optimality, e.g. $1-(1-\eta^t)^L = \Theta(1)$, $nLs'^t = o(n)$