Probability

1. Event:
$A \subseteq \Omega \;\; s.t. \;\; Pr_p[A] = \Sigma_{x \in A} p(x)$

2. Independent:
A and B independent iff $Pr[A \cap B] = Pr[A] \cdot Pr[B]$

3. Law of Total Probability:
$Pr[B] = \Sigma_{i=1}^{n} Pr[B \cap A]$

4. Conditional Probability:
$Pr[A|B] = \frac{Pr[A \cap B]}{Pr[B]}$

5. Union Bound:
$Pr[\bigcup_{i=1}^{n} A_i] \leq \Sigma_{i=1}^{n} Pr[A]$

6. Expectation:
$Ex[x] = \Sigma_{i=1}^{n} Pr[X = x_i] \cdot x_i$

7. Linearity of Expectation:
$Ex[x] = Ex[\Sigma_{i=1}^{n} x_i] = \Sigma_{i=1}^{n} Ex[x_i]$

8. Multiplicative Expectation under Independence:
$Ex[XY] = Ex[X] \cdot Ex[Y]$

9. Variance:
$Var[X] = \sigma^2 = Ex[X^2] - E[X]^2$
$Var[cX] = c^2 Var[X]$

10. Linearity of Variance under Pairwise Independence:
$Var[\Sigma_{i=1}^{n} X_i = \Sigma_{i=1}^{n} Var[X_i]$

11. Markov's Inequality:
$Pr[Y \geq a] \leq \frac{Ex[Y]}{a}$

12. Chebyshev's inequality:
$Pr[|X - \mu| \geq k] \leq \frac{\sigma^2}{k^2}$

13. Chernoff Bound:
$Pr[X > (1 + \beta)\mu] < e^{-\beta^2 \frac{\mu}{2}}$,
$\beta \in (0, 1)$
$Pr[X > (1 + \beta)\mu] < e^{-\beta \frac{\mu}{3}}$,
$\beta > 1$
$Pr[X < (1 - \beta)\mu] < e^{-\beta^2 \frac{\mu}{2}}$,
$\beta \in (0, 1)$

Master Theorem

Leaf: $f(n) = O(n^{log_b a - \varepsilon}),\ \varepsilon > 0$
$\rightarrow T(n) = \Theta(n^{log_b a})$

Balance: $f(n) = \Theta(n^{log_b a} log^k n),\ k \geq 0$
$\rightarrow T(n) = \Theta(n^{log_b a} log^{k+1} n)$

Root: $f(n) = \Omega(n^{log_b a + \varepsilon}),\ \varepsilon > 0$
$\bigcap af(\frac{n}{b}) < cf(n),\ c \in (0, 1)$
$\rightarrow T(n) = \Theta(f(n))$

Amortized Analysis

1. Accounting Method:
Follow how many ops per element in each step

2. Potential Method:
   (a) Goal: Potential method makes $\hat{c} = O(1)$
   (b) D_i is the state of the data structure after the i-th step.
   (c) $\Phi(D_0) = 0$
   (d) $\Phi(D_i) \geq 0$ for all $i \in \mathbb{Z}$
   (e) $\hat{c_i} = c_i + \Phi(D_i) - \Phi(D_{i-1})$,
   (f) $\Sigma_{i=1}^{n} \hat{c_i} = \Sigma_{i=1}^{n} c_i + \Phi(D_i) - \Phi(D_{i-1})$

3. Competitive Analysis:
Can use Potential Function. Online algorithm A is α-competitive if $\exists$ a constant k such that, for any sequence of inputs S:
$$C_A(S) \leq \alpha C_{opt}(S) + k$$

Union Find

1. 3 Operations
   (a) Make-Set(a): Make a set of size 1 with a
   (b) Find-Set(a): Find root of tree with a
   (c) Union(a, b): Find roots of both and make one root point to the other

2. Union by Rank: $O(mlogn)$
$rank \geq height$. Node with smaller rank always changes its pointer. If ranks are equal, the one who got pointed to got rank increased by 1. At any point, height is at most $logn$.

3. Path Compression: $O(n + f \cdot (1 + log_{2+f/n} n$
During Find-Set, does not change rank. For each node transversed, change parent pointer to the root found.

4. Both Together: $O(m\alpha(n))$

5. Inverse Ackermann Function:
$\alpha(n < 10^{80}) \leq 4$

Hashing

1. Hash Function:
$m =$ array size
$K \subseteq U$, set of keys in Universe
$|K| = n$
T is array of size m
$h :=$ function that maps universe, U, of all possible keys to buckets $\{0, .., m - 1\}$
$h : U \rightarrow \{0, ..., m - 1\}$

2. SUHA:
each key $k \in K$ equally likely to be hashed to any bucket in T. Leads to a load factor α of n/m.

3. Chaining
Linked list collision resolution

4. Universal:
A set of Hash functions $\mathcal{H}$ is universal if $\forall x, y \in U$, where $x = y$:
$Pr_{h \in \mathcal{H}}\{h(x) = h(y)\} = \frac{1}{m}$

5. Theorem:
Given a Universal $\mathcal{H}$, for any x:
Ex[num. of collisions with x] $< \frac{n}{m} = \alpha$

6. Universal Hash Function Ex:
m is prime. Key k decomposed into $r + 1$ digits each with value $\in [0, m)$. $k = \Sigma_{i=0}^{r} k_i$
Pick $a = (a_0, .., a - r)$ where each a_i is assigned a distinct value from $[0, m)$.
$h_a(k) = \Sigma_{i=0}^{r} a_i \cdot k_i \text{mod} m$ $\mathcal{H} = \{h_a : \forall a\}$

7. Perfect Hashing:
Hashing with no collisions, table is static.
Ex: Table size $m = n$. Hash everything. Sub-bucket size = collisions^2. Random hash from $\mathcal{H}$ until no collisions.
$Ex[\text{collisions}] = \frac{n^2}{m} = O(n)$

8. Open Addressing:
$h : Ux\{1, ..., m\} \rightarrow \{0, ..., m - 1\}$

9. Probe Sequence:
$h(k, 1), ..., h(k, |\mathcal{H}|)$
if $h(k, i)$ collided, try $h(k, i + 1)$

10. Open Addressing modification:
When deleted, mark as deleted, searching ignores this flag and pretends it is full, insert inserts into deleted bucket.

11. Uniform Hashing Assumption:
Each probe in probe sequence maps to a random bucket uniformly. Probes at most $\frac{1}{1-\alpha}$.

MST Trees

1. Minimum Spanning Tree:
   Given graph G, find tree with min weight.
2. Cut:
   For disjoint subgraphs A and B, cut is all edges between them.
3. The Cut Property:
   min edge in cut always in some MST.
4. Kruskal's Algorithm: $O(ElogV)$
   Init with empty T. Examine edges in increasing weight order. If edge connects two unconnected components, union them and add that edge. Else, discard that edge (cycle). Terminate when $|V| = n$.
5. Prim's Algorithm: $O(E + VlogV)$
   Start with $T = \{r\}$, r vertex. Add min edge from cut T and $G \backslash T$ to some vertex $\notin T$

Linear Programming

1. Standard LP form:
   $A \in \mathbb{R}^{m \times n}.b, c \in \mathbb{R}^n$
   max $c^T x$
   s.t. $Ax \leq b$
   $x \geq 0$
2. LP Duality:

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

3. Weak LP Duality:
   if x, y feasible:
   $\rightarrow c^T x \leq b^T y$
   $c^T x \leq y^T Ax \leq y^T b$
4. Strong LP Duality:
   Both feasible:
   $c^T x^* = by^*$

   LHS unbounded:
   RHS infeasible

   LHS infeasible:
   RHS unbounded

   Both infeasible:

Max Flow

1. Real Flow:
   $\tilde{f}$
2. Net Flow:
   $f(u, v) = \tilde{f}(u, v)$
   $f(v, u) = -\tilde{f}(u, v)$
3. Capacity Constraint:
   $f(u, v) \leq c(u, v)$
4. lemma:
   if $x \cap y = \emptyset$,
   $f(x \cup y, z) = f(x, z) + f(y, z)$
5. Residual Network:
   Let $c_f(u, v) = c(u, v) - f(u, v)$
   $G_f(V, E_f)$ such that:
   (a) if $c_f(u, v) > 0$
   $\rightarrow (u, v) \in E_f$
   (b) if $c_f(v, u) > 0$
   $\rightarrow (v, u) \in E_f$
   (c) $w((u, v)) = c_f(u, v), \forall (u, v) \in E_f$
6. Augmentation Path:
   Any path p from s to t in G_f.
   Flow can be increased by:
   $c_f(p) = min_{(u,v) \in p}\{c_f(u, v)\}$
7. Max Flow, Min-Cut:
   (a) $|f| = c(S, T)$ for some cut (S, T)
   (b) $f = f^*$ is max flow
   (c) f has no agumentation paths.
8. Ford-Fulkerson: $O(f^* E)$
   Keep augmenting until you can't anymore.
9. Edmonds-Karp: (VE^2)
   Augment along shortest path.
   $O(VE)$ augmentations. $O(E)$ BFS for shortest path.

Game Theory

1. Nash Equilibrium:
   There exists a collection of randomized strategies such that no player has no incentive to change their own strategy.
2. Setting:
   Only for 2 player, 0-sum games.
   Given $m \times n$ utility matrix M,
   $u_A(x, y) = \sum_{i=1}^m \sum_{j=1}^n M[i, j] x_i y_j$
   $u_B(x, y) = -u_A(x, y)$

Gradient Descent

1. β-Smooth:
   f is β-smooth iff:
   $|f(x + \delta) - [f(x) - (\nabla f(x))^T \delta]| \leq \frac{\beta}{2} ||\delta||_2^2$
   $x^T \mathcal{H} x \leq \beta$, unit norm x.
2. $||x||_2 = \sqrt{\sum_{i=1}^n x_i^2}$
3. Taylor Approximation:
   $f(x + \delta) \approx f(x) + (\nabla f(x))^T \delta + \frac{(\nabla^2 f(x))^T}{2!} \delta^2 + ... + \frac{(\nabla^n f(x))^T}{n!} \delta^n$
4. Proposition:
   if f β-smooth then:
   $f(x^{(x+1)}) \leq$
   $f(x^{(t)}) - (\eta_t - \frac{\beta}{2} \eta_t^2) ||\nabla f(x^{(t)})||_2^2$
   $\eta_t < \frac{2}{\beta}$
5. Convex:
   $f(\lambda x + (1 - \lambda) y) \leq$
   $\lambda f(x) + (1 - \lambda) f(y)$,
   $\forall x, y \in \mathbb{R}^n, \lambda \in [0, 1]$;
   $f(x + \delta) \geq f(x) + (\nabla f(x))^T \delta, \forall x, \delta$
6. α-Strong Convexity:
   f is α-strongly convex iff:
   $f(x + \delta) - f(x) - (\nabla f(x))^T \delta \geq \frac{\alpha}{2} ||\delta||_2^2$, $\forall x, \delta$
   $x^T \mathcal{H} x \geq \alpha$, unit norm x
7. Claim:
   if f convex and differentiable, all critical points global minima
8. Proposition:
   f α-strongly convex and β-smooth, T w/ $\eta = \frac{1}{\beta}$ for:
   $f(x^{(T)}) \leq f(x^*) + \varepsilon$
   $T \geq 4 \frac{\beta}{\alpha} log \frac{f(x^(0)) - f(x^*)}{\varepsilon}$

Game Theory Cont.

1. Gedanken Experiment:
   If row player had to commit to randomized strategy, they would solve:
   $max_{x \in \delta_m} min_{y \in \delta_n} x^T R y$
   $\delta_m = \{x \in \mathcal{R}_{\geq 0}^m | \sum_i x_i = 1\}$
   $x^T R y = \sum_{ij} R_{ij} x_i y_j$, expected payoff for row player.
2. Min-Max Theorem:
   $max_{x \in \delta_m} min_{y \in \delta_n} x^T R y = min_{y \in \delta_n} max_{x \in \delta_m} x^T R y$

max z	minz'
s.t. $x^T R \geq z 1^T$	s.t. $-y^T R^T +$
$x^T 1 = 1$	$z' 1^T \geq 0$
$x_1 \geq 0$	$y^T 1 = 1$
	$y_j \geq 0$

Intractability

Def: P - Solvable in poly time

Def: NP - Polynomial verification

Def: NP-Hard - If it is at least as hard as any problem in NP

Def: NP-Complete - In NP and NP-Hard

Reduction

$\pi_1 \leq_p \pi_2$ (π_2 is at least as hard as π_1) iff

- if x is input to π_1, then $R(x)$ is an input to π_2
- $\pi_1(x) = \text{yes} \iff \pi_2(R(x)) = \text{yes}$

If $\pi_1 \leq_p \pi_2$ and $A()$ is poly alg for π_2, then $A(R())$ is a poly alg for π_1

NP-Complete

Circuit-Sat

Circuit of AND, OR, NOT. Does there exist an input of 1?

SAT

Boolean formula. Create formula from C-SAT.

3-SAT

3 things OR and then AND all the clusters. Subset of SAT.

Vertex Cover

Does there exist $S \subseteq V$ such that $|S| \leq k$ and every edge incident to one vertex in S. 3-SAT to graph.

BIN-ILP

Given linear program, solution where each variable is binary? From 3-SAT converting clauses into constraints.

Clique

Does there exist an induced graph that is complete? Create complement graph and find vertex cover.

Dominating Set

Every vertex adjacent to S. Construct new graph where edge is vertex incident to its vertices and find a vertex cover.

Approximation

Minimum V-C

Pick any edge (u, v), add to C. Remove every edge incident to u or v. 2-approx

Set Cover

X union of many sets. Find min num sets such that union equals X. $ln(m)$ approximation via greedy.

proof

$k = |C_{opt}|$. Every added set adds at least $1/k$ amount to solution. u not covered elements. $u_{i+1} = u_i(1 - 1/k)$. Algebra.

V-C Int-Lin

constraint $1 \leq x_i + x_j$ for every edge $(i, j) \in E$. if $x_i = 1$, its in V-C. $x_i \in [0, 1]$. If $x_i \geq 1/2$, then add to V-C. 2-Approx via math.

TSP

Compute MST. Walk by visiting each edge twice. Create into cycle via shortcuts, when repeat vertex, simply skip it in cycle construction. 2 approx as walk 2x and cycle less or equal to that.
Another method, construct euler tour of only odd vertices. Use half of it and it costs at most half opt. Use that and original walk, shortcut, and get 3/2 approx.

Maximal Matching

Add edge, remove all adjacent edges. 2-approx because each edge blocks at most 2 other optimal edges.

MAX-k-SAT

Max number of 1 clusters. Random and math for $\frac{1}{1-2^{-k}}$-approx.

Bin-Packing

Fit items into buckets using min buckets. Greedy. 2-approx as upper-limit on half-filled buckets.

Max Cardinality matching

Set of edges such that no edges share an endpoint. Kinda like maximal matching.

Random Walks

Def: $d(u) = \sum_{e \in E | e = (u,v)} w_e$

Def: p_v^t - prob walk visits v at step t. $p_v^{t+1} = \sum_{e \in E} \frac{w_e}{d(u)} p_u^t$

Def: $p^t = [p_1^t ... p_n^t]^T$

Def: Lazy Random Walk - Random walk but with $p_l azy$ probability of staying still.

Def: Adjacency Matrix - $A_{u,v} = w_e if(v, u) \in E$

Def: Degree Matrix - $D_{u,v} = d(u) if u = v$.

Def: Walk Matrix - $W = AD^-1.W_{u,v} = \frac{1}{d(u)} if(v, u) \in E$. $\hat{W} = P_{lazy} I + (1 - P_{lazy})W$. $p^t = W^t p^0$

Def: Stationary Dist. - $W\pi = \pi$

Def: aperiodic - it is not the case that every cycle is a multiple of c ¿ 1

Theorem

Every connected non-bipartite undirected graph has a unique stationary distribution which random walks are guaranteed to converge. If lazy, no bipartite. If directed, graph is strongly connected and aperiodic.

Sketching and Similarity

$S := \{[n] | Pr[i \in S] = 1/r\}$. Find x_s by setting x_i to 0 if $\notin S$ (i.e. only indices in S checked). $x_s = y_s$ with prob $(1 - 1/r)^{D(x,y)}$. Prob $f(x_s) = f(y_s)$ between $e^{-D(x,y)/r} - .01 < (1 - 1/r)^{D(x,y)} \leq e^{-D(x,y)/r}$.

Streaming and Sublinear

Majority Element

if maj-el = el, counter + 1
elif counter = 0, maj-el = el, counter = 1
else counter - 1.
Maj-el appears at most 1/2 before every counter 0, therefore appears more than 1/2 at final.

Random sample stream

r = i_j with prob $1/j$. Proof by induction.

Count distinct elements

Simple, large if $DE > 4r$, small if $DE < r/4$. Hash function to [r] uniform. If any hash to 0, report large, otherwise small.

Approx diameter of set

Pick one vertex randomly. Find other to maximize diameter. O(n). At least diameter/2.