さすが

Algorithm	**Intuition**	**Bounds (T runtime, S space)**
QuickSelect	Randomized quicksort on half	$T(n) = \theta(n)$
DetSelect	Median of medians, $\lceil \frac{g}{2} \rceil$ have $\lceil \frac{gsize}{2} \rceil$ elements $\leq$ or $\geq$ the pivot	$T(n) \leq O(n)$
Kruskal MST	Sort edges, greedily pick avoiding cycle by using union-find	$T(n) \leq O(m \log m)$ sort + UnionFind
Prim MST	All edges into minheap, pick starting vertex, repeatedly add shortest edge	$T(n) \leq O(m \log n)$ pq $T(n) \leq O(m + n \log n)$ fibheap
Union Find	Trees for connected components, union by rank, lazy path compression	$T(n) \leq O(\log n)$ worst F/U $T(n) \leq O(\log^* n)$ amortized F/U
Perfect Hashing	Method 1. Univ hash with $O(n^2)$ space Method 2. Univ hash into $O(n)$ table, then $O(n)$ table (squaring sizes) per bucket	M1. $S(n) = O(n^2)$ M2. $S(n) = O(n)$, $E[\Sigma_i (L_i)^2] < 2n$ indicators to $P[\Sigma_i (L_i)^2 > 4n] < \frac{1}{2}$
Majority	Counter, different gangs shoot each other	
ε-heavy hitter	Count top k elements, decrement all buckets by 1 upon non-heavy hitter, $0 \leq c_t(e) - e_t(e) \leq \frac{t}{k+1} \leq \varepsilon t$, $k = \lceil \frac{1}{\varepsilon} \rceil - 1$	$S(n) = (\log \Sigma + \log t)\ O(\frac{1}{\varepsilon})$ (elem, cnt) * (num elem)
ε-heavy hitter with deletions Misra-Gries	Idea: use hashtable for counts, error is $e_t(e) - c_t(e) = \Sigma_{e \neq e'} c_t(e')\ 1(h(e) = h(e'))$ and $E[error] \leq \frac{\lvert S_t \rvert}{k} = \frac{\lvert active\ set_t \rvert}{num\ hashtable\ slots}$ Apply boosting to reduce error, m hashtables each with their own counts, by Markov $P\{error > 2\frac{\lvert S_t \rvert}{k}\} \leq \frac{1}{2}$ hence $P\{all\ large\ error\} \leq \frac{1}{2^m}$, so $P\{min\ of\ ests\ is\ small\ error\} = 1 - \frac{1}{2^m}$ Picking $k = \frac{2}{\varepsilon}$ and $m = \lg \frac{1}{\delta}$ we get $P\{\lvert best_t(e) - count_t(e) \rvert \leq \varepsilon \lvert S_t \rvert\} \geq 1 - \delta$	$(\lg k)(\lg \lvert\Sigma\rvert)$ bits per hash fn k counters of at most $\lg t$ bits For boosted final version, $S(n) = O(\frac{1}{\varepsilon} \lg \frac{1}{\delta})$ km counters + $(\lg \frac{1}{\varepsilon})(\lg \lvert\Sigma\rvert) O(\lg \frac{1}{\delta})$ m hash fns TLDR: $\frac{1}{\delta}$ times polylog factors
String Equality	Draw p from $[1,\ M = 2sN \lg(sN)]$ Send $p,\ x \bmod p$, recv $x \bmod p =?\ y \bmod p$ $2^N \geq D = \lvert y - x \rvert = p_1^{i_1} \ldots p_k^{i_k} \geq 2^k$ so $P\{false\ positive\} \leq \frac{N}{\pi(M)} \leq \frac{1}{s}$	$S(n) \leq 2 \lg M$ send $p,\ x \bmod p$ i.e. $S(n) = O(\log N)$
Karp-Rabin	$m = \lvert text \rvert, n = \lvert pat \rvert$, pick $p \in [1, M = 2sn \lg(sn)]$, store $h_p(P),\ h_p(2^n)$, rolling hash in constant time $h_p(x') = (2h_p(x) - x_{hb} h_p(2^n) + x'_{lb}) \bmod p$	$T(n) = O(m + n)$ initial hash is $O(n)$, following hash $O(1)$ so $O(m)$ of those. $O(\log m + \log n)$ bits for p if $s = 100m$

<u>LCS</u> LCS_{mn} longest $O(mn)$ 0 $\quad if\ i = 0\ or\ j = 0$ $max\{LCS_{i-1,j}, LCS_{i,j-1}\}\ if\ S_i \neq T_j$ $1 + LCS_{i-1,j-1}$ $\quad if\ S_i = T_j$	<u>Knapsack</u> $V(n, S)$ highest value $O(nS)$ 0 $\quad if\ k = 0$ $V(k-1, B)$ $\quad if\ s_k > B$ $max\{v_k + V(k-1, B - s_k), V(k-1, B)\}\ else$	<u>MWIS (tree)</u> $max\{U(r), N(r)\}$ $O(n)$ indep set = no edge both endpt in set use $U(v) = w_v + \Sigma_{u \in C(v)} N(u)$ not use $N(v) = \Sigma_{u \in C(v)} max\{N(u), U(u)\}$

さすが

OBST $C_{1,n}$ $O(n^3)$ / $O(n^2)$

0 $if\ i > j$

f_i $if\ i = j$

$\min_{i \le k \le j} f_{i,j} + C_{i,k-1} + C_{k+1,j}\ else$

where $f_{i,j} = \sum_{k=i}^{j} f_k$

Dijkstra SSSP $O(m \log n)$ or $O(m + n \log n)$

while nodes unvisited: visit cheapest n, update neighbor costs to $min(old,\ n + edge)$

Can't do negative edges because we don't revisit visited nodes.

Bellman-Ford SSSP $D_{v,\,n-1}$ $O(mn)$

0 $if\ k = 0\ and\ v = s$

∞ $if\ k = 0\ and\ v \ne s$

$min\{D_{v,\,k-1},\ \min_{x \in N(v)} D_{x,\,k-1} + len(x, v)\}\ else$

"Extend one edge at a time"

Can detect negative cycles.

Matrix APSP $O(n^3 \log n)$

$B_{ij} = min_k\{A_{ik} + A_{kj}\}$ (≤ 2 edges)

$C = B \times B$ (≤ 4 edges), ...

$O(\log n)$ squarings, mult is $O(n^3)$

Floyd-Warshall APSP $O(n^3)$

for k in [1,n]: forall i,j:

$A_{ij} = min\{A_{ij}, A_{ik} + A_{kj}\}$

Johnson APSP $O(mn + n^2 \log n)$

Add dummy node with len 0 to every other, run Bellman to find shortest path, add that length of shortest path to all so that nonnegative, run Dijkstra from every node

TSP $T(n) = O(n^2 2^n),\ S(n) = O(n 2^n)$

$len(x, t)$ $if\ S = \{x, t\}$

$\min_{t' \in S,\, t' \ne t,\, t' \ne x} C(S - t,\ t') + len(t',\ t)\ \ else$

MATH

- Prime Number Theorem: $\lim_{n\to\infty} \frac{\pi(n)}{n / \ln n} = 1$ $P\{x\ is\ prime \mid x \in [n]\} \ge \frac{1}{\ln n}$ for $k \ge 4,\ n \ge 2k \lg k \Rightarrow \pi(n) \ge k$
- $\ln(n+1) < H_n < 1 + \ln n$, $H_n \approx \log n$, $(\frac{n}{k})^k \le \binom{n}{k} \le (\frac{ne}{k})^k$, $\frac{x}{\ln x - 1} < \pi(x) < \frac{x}{\ln x - 1.1}$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)}$: $0 \le f \in O(g) < f \in o(g)$; $(0, \infty) = f \in \theta(g)$; $\infty > f \in \omega(g) \ge f \in \Omega(g)$
- $\sum a_1 r^k = \frac{a_1}{1-r}$, $\sum_{k=0}^{n} r^k = \frac{1-r^{n+1}}{1-r}$, $\sum_{k=0}^{n} k r^k = \frac{r(n\, r^{n+1} - (n+1)\, r^n + 1)}{(r-1)^2}$, $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, $\sum_{i=1}^{n} i^2 = \frac{1}{6} n(n+1)(2n+1)$, $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$
- Universal: $\forall x \ne y\ P_{h \leftarrow H}\{h(x) = h(y)\} \le \frac{1}{M}$
- k-Universal: $\forall x_1 \dots x_k$ distinct, v_1 ... v_k anything, $P_{h \leftarrow H}\{\wedge\, h(x_i) = v_i\} = \frac{1}{M^k}$
- k-Universal $\Rightarrow$ m-Universal $\forall\ 1 \le m \le k$

THEOREMS

- $T(n) = a\, T(\frac{n}{b}) + cn^k$ solves to $\theta(n^k)$ if $a < b^k$, $\theta(n^k \log n)$ if $a = b^k$, $\theta(n^{\log_b a})$ if $a > b^k$
- $T(n) \le T(a_1 n) + T(a_2 n) + \dots + T(a_k n) + cn$ with $\Sigma a_i < 1$ implies $T(n) \in O(n)$
- All comparison-based sorts need at least $\lg(n!)$ compares to sort n elements
- $Amortized\ Cost\ :=\ Actual\ Cost + \Delta\Phi$, i.e. $A_i = c_i + \Phi(s_i) - \Phi(s_{i-1})$

(Potential Function) If the heights of the stacks are a and b, there are at least $3|a - b|$ tokens in the system. For each unit-cost operation, the potential increases by at most 3, and the actual cost is 1, so the amortized cost is at most 4. For a rebalance with k elements, the potential changes from $3k$ to at most 3, and the actual cost is at most $3k + 1$, so the amortized cost is at most $\Delta\Phi + c_i = (3 - 3k) + (3k + 1) = 4$. Hence the proof.

Discrete	$E[X]$	$Var[X]$	$p_x(x)$	
$Bernoulli(p)$	p	pq	p at 1 , q at 0	
$Binomial(n,\ p)$	np	npq	$\binom{n}{x} p^x q^{n-x}$	
$Geometric(p)$	$\frac{1}{p}$	$\frac{q}{p^2}$	$q^{x-1} p$	
$Poisson(\lambda)$	λ	λ	$\frac{e^{-\lambda} \lambda^x}{x!}$	
Continuous	$E[X]$	$Var[X]$	$f_x(x)$	$F_x(x)$
$Uniform(a, b)$	$\frac{b+a}{2}$	$\frac{(b-a)^2}{12}$	$\frac{1}{b-a}$ $a \le x \le b$ 0 else	$\frac{x-a}{b-a}$ $x \in [a, b)$ $0\ if\ x < a$ $1\ if\ x > b$
$Exponential(\lambda)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\lambda e^{-\lambda x}$ $x \ge 0$ 0 else	$F_x(x) = 1 - e^{-\lambda x}$ $x \ge 0$ $F_x(x) = 0$ else
$Normal(\mu,\ \sigma^2)$	μ	σ^2	$\frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{1}{2} (\frac{x-\mu}{\sigma})^2)$	

$E[X] = \Sigma_x\, x\, P\{X = x\}$

$E[g(X)] = \Sigma_x\, g(x)\, P\{X = x\}$

$Var[X] = E[(X - E[X])^2]$

$Var[X] = E[X^2] - E[X]^2$

$E[X] = \int_{-\infty}^{\infty} x f_x(x)\, dx$

$Var[X] = \int_{-\infty}^{\infty} (x - E[X])^2 f_x(x)\, dx$

Markov

$P\{X \ge a\} \le \frac{E[X]}{a}$

Chebyshev

$P\{|X - \mu| \ge k\sigma\} \le \frac{1}{k^2}$

Jensen

$F(E[X]) \le E[F(X)]$ for X rv, F convex