Asymptotics

$O(\cdot)$	$\le$	$\lim_{n\to\infty} \frac{F}{G} < \infty$
$\Theta(\cdot)$	$=$	$= C > 0$
$\Omega(\cdot)$	$\ge$	$\downarrow \; = 0$

n
n/b
n/b^k
1

Depth: $\log_b n = k$

Width: $a^{\log_b n} = n^{\log_b a}$

c
$< \log^*$
$< \log$
$< n^{1/3}$
$< n^{1/2} = \sqrt{n}$
$< 2^{\log n} = n$
$< n\log(n)$
$= \log(n!)$
$< n^2$
$< 2^{\sqrt{n}}$
$< 2^n$
$< n!$

$T(n) = a \cdot T(n/b) + C \cdot n^d$

↳ Work @ level k: $a^k \cdot O\left(\frac{n}{b^k}\right)^d = O(n^d) \cdot \left(\frac{a}{b^d}\right)^d$

Total Work:

↳ $n^d \sum_{i=0}^{k} (a/b^d)$

$$\therefore T(n) = \begin{cases} \Theta(n^d) & : a/b^d < 1 \equiv \log_b a < d \\ \Theta(n^d \log(n)) & : a/b^d = 1 \equiv \log_b a = d \\ \Theta(n^{\log_b a}) & : a/b^d > 1 \equiv \log_b a > d \end{cases}$$

$n! < n^n$ | $n! > \left(\frac{n}{2}\right)^{n/2}$

$\log(b) < \log(b)$ | $\log(b) > \log\left(\frac{n}{2}\right)^{n/2}$

$= O(n \log n)$ | $= \Omega\left(\frac{n}{2}\log\left(\frac{n}{2}\right)\right)$

$= \Omega(n \log n)$

$n^{\log n} = (2^{\log_2 n})^{\log n} = 2^{\log n \log n}$

- $\log_b N = \frac{\log_b N}{\log_a b} \Rightarrow \log_a N = \log_b N \cdot \log_a b$
- $x = b^{\log_b x}$ | $\log_b\left(\frac{x}{y}\right) = \log_b x - \log_b y$

$a(1 + p + p^2 + \dots p^k) = a\sum_{i=0}^{n} p^i = a(p^{k+1} - 1/p-1)$ | $1 + x \le e^x$

→ Bits in binary representation of N: $\lceil \log(N+1) \rceil$

→ Depth of binary tree w/ N-nodes: $\lfloor \log(N) \rfloor$

→ Fast exp w/repeated squaring: $9^{71} = 9^{64} \cdot 9^4 \cdot 9^2 \cdot 9^1$

[sum of 3 digits] $\le$ [2-digit num]

$\underbrace{x_b + y_b + z_b}_{k\text{-digits in base-}b} \le \alpha \cdot b + \gamma \Rightarrow 3(b-1) \le b^2 - 1$

span $[0, b^k) \equiv [0, b^k - 1]$

[len base-2] $\le 4 \times$ [len base-10]

$\lceil \log(x+1) \rceil \le 4 \cdot \lceil \log_{10}(x+1) \rceil$

$\log 10^n \le 4 \cdot \log_{10} 10^n = 4n$

$2^{\log 10^n} \le 2^{4n}$

$10^n \le 16^n$

D-ary tree w/ n-nodes depth:

$n = \sum_{i=0}^{f} d^i = (d^{f+1} - 1)/(d-1)$

$nd - n = d^{f+1} - 1$

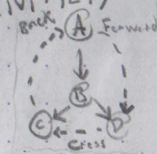
$\log_d(n) + \log_d(d-1) = \le \log_d(d^{f+1} - 1)$

$\log_d(n) + 1 + \log_d(1) = \le (f+1)$

$\Rightarrow f = \lfloor \log_d n \rfloor = \Omega\frac{\log n}{\log d}$

DFS

Stack, FIFO : $O(|V| + |E|)$



Edge Types, $e := (u, v)$

1.) Tree: $e \in E'$
- u is parent of v
- e traversed in DFS

2.) Back: $e \notin E'$

→ u descendent of v in DFS

→ Visit v prior ⇒ cycle

3.) Forward

→ u ancestor of v in DFS

→ skipped e, $\exists w_{e'} < w_e$

4.) Cross

→ u, v are cousins

→ Different, disjoint DFS calls

Tree/Forward:

$[\ [\ ]\ ]$
$u\ v\ v\ u$

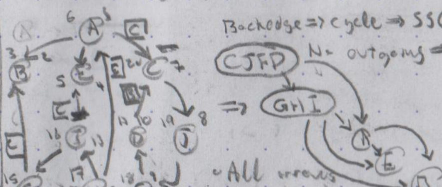
Backs: $(\ [\ ]\ )$
$v\ u\ u\ v$ | $post_u < post_v$

Cross:

$(\ )\ [\ ]$
$v\ v\ u\ u$

$[\ ]\ (\ )$
$u\ u\ v\ v$

- Source: Highest post, visit first
  → No incoming edges from other SCCs
  → Linearize by popping source
- Sink: Lowest post, visit last
  → No outgoing edges to other SCCs



- All arrows one direction
- Ordered decreasing post

→ Process paths in increasing weight

Dijkstra's: Find shortest path between one node and all others [non-neg weights]

$O((|V| + |E|)\lg(|V|))$

→ Shortest path $s \to t$ through v_0: $O(|V| + |E|\log|V|)$

Find $v_0 \to t$ on G and $t \to s$ on G^R

Bellman-Ford: Shortest path, w/neg weights

$O(|V| \cdot |E|)$ by iterating $|V| - 1$ times

BFS: Shortest path w/no weights

$O(|V| + |E|)$ → Fewest edges, unlike DFS (e.x. complete)

List, $|V|^2$ for matrix

D&C

1.) Break into 'a' subproblems of size $\frac{n}{b}$
2.) Recurse until base-case
3.) Solve with $c \cdot n^d$ work

Ex: Karatsuba's:

$x = 10^{n/2} \cdot x_H + x_L$

$A = x_H \cdot y_H$

$B = x_L \cdot y_L$

$D = (x_H + x_L) \cdot (y_H + y_L)$

$x \cdot y = A \cdot 10^n + (D - A - B) \cdot 10^{n/2} + B$

$\Rightarrow T(n) = 3T\left(\frac{n}{2}\right) + O(n) = O(n^{\log_2 3})$

- IF C, C' are SCCs, and edge $C \to C'$
  ⇒ Highest post in C > Highest in C'

SSC Algo

I.) DFS on G^R to Find $\overbrace{\text{Sink(s)}}^{\text{sources of } G}$

2.) DFS on G w/ decreasing postorder From (1)

MST

Tree: (connected, acyclic) ⇔ $|E| = |V| - 1$

↳ Removing cycle edge keeps connected

↳ Tree iff unique (one) path between all nodes

- Cheapest edge across cut must be in MST

MOST PROB | Device outputs ints $\in [n]$ w/chance p_i: Find most likely sum of 4:

$P_r[x_1 + x_2 + x_3 + x_4 = n] = P[x_1 + x_2 \dots]$

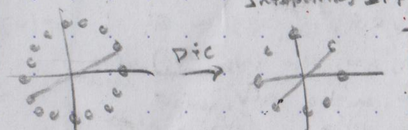
$A(x) := \sum_{i=1}^{n} p_i x^i$, $B(x) := A(x)^4$

Highest coeff of B = most likely

FFT

$[a_0, a_1, \dots a_d]$ $\xrightarrow{\text{Eval, FFT}}$ $[A(x_0), A(x_1) \dots A(x_{d-1})]$

Interpolation, IFFT



$e^{\frac{2\pi i}{n}}$

$\omega^{j + n/2} = -\omega^j$

$\omega \neq 1 : \sum_{k=0}^{n-1} \omega^k = \frac{\omega^n - 1}{\omega - 1} = \frac{1 - 1}{\omega - 1} = 0$

$\omega^4 = -1$

$\omega^8 = (-1)^2$

$e^{\frac{2\pi i \cdot k}{8}} = (-1)^2$ ... $\forall k \in [1, 8]$

$e^{\frac{2\pi i k}{4}} = -1$

$\Rightarrow k = \{1, 3, 5, 7\}$

Zero-pad

$n = 2^{\lceil \log(d_a + d_b + 1) \rceil}$

↳ C has max deg. 2d

⇒ eval @ 2d+1 pts

Pad A: $n - d_a$

Pad B: $n - d_b$

TRI-SUM

Given $|A| = n$, $A_i \in [0, n]$, Find i, j, k st $A_i + A_j + A_k = n$

$p(x) := \sum_{i=0}^{n-1} x^{A_i} = x^{A_0} + x^{A_1} + \dots + x^{A_{n-1}}$

$q(x) := p(x)^3 = \sum_{i,j,k \in [0,n]} x^{A_i + A_j + A_k}$

↳ Coeff on x^n ∝ how many elts sum to n

Cartesian Sum

Let A, B be sets of integers $\in [0, 10n]$

Find $A + B := \{a + b \mid a \in A, b \in B\}$

↳ $\{1, 3\} + \{2, 4\} = \{3, 5, 7\}$

$P := \sum_{a \in A} x^a$, $Q := \sum_{b \in B} x^b$

$R := P \cdot Q = r_0 + r_1 x^1 + \dots r_{20n} x^{20n}$

↳ $r_k = \sum_{j=0}^{k} p_j \cdot q_{k-j}$

∴ r_k counts how many $a + b = k$

⇒ cartesian sum ∝ non-zero r_k

CROSS-COR | $\underbrace{g \in \{0,1\}^n}_{\text{pattern}}$, $\underbrace{s \in \{0,1\}^m}_{\text{sequence}}$

k-matches

$p_1(x) = \sum_{i=0}^{n-1} a_i x^i$

↳ $a_j = g'(n - j - 1)$ → [Reversed!]

$p_2(x) = \sum_{i=0}^{m-1} b_i x^i \to b_j = s'(j)$

$p_3(x) = p_1 \times p_2$ st coef of x^{n-1+j}

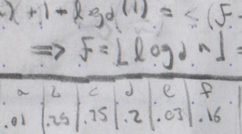
$c_{n-1+j} = \sum_{i=0}^{n-1} a_{n-1-i} \cdot b_{j+i}$ $\forall j \in [0, m-n]$

$= \sum_{i=0}^{n-1} g'(i) \cdot s'(j+i)$

→ out j's st $c_{n-1+j} \ge n - 2k$

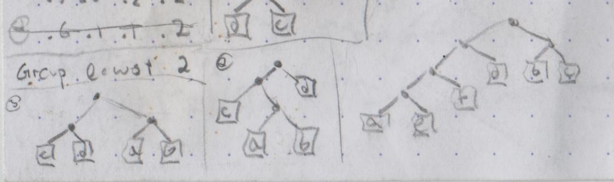
Huffman

	a	b	c	d
①	.25	.5	.15	.1
②	.1	.1	.1	.7
③	.3	.3	.2	.2
~~④~~	~~.6~~	~~.1~~	~~.1~~	~~.2~~



a	b	c	d	e	f
.01	.25	.35	.2	.03	.16

Group lowest 2



- Doesn't always work/compress
  → All same freq. ⇒ binary tree
- Longest Codeword len is n-1
  → IF Freqs are $\frac{1}{2}, \frac{1}{4} \dots \frac{1}{2^{n-2}}, \frac{1}{2^{n-1}}, \frac{1}{2^{n-1}}$
- Created in $O(n \log n)$
  (sort frequencies dominates

GUESS-STRAT

Given w_i is chosen w/ prob. p_i $\forall i \in [20]$

Find best strat to guess correct w_i:

→ Guesses ∝ Code length

→ min guesses ∝ min code length

↳ what Huffman does!

Op	Matrix	List
Size	n^2 bits	$O(m+n)$ words
$\exists e : (u,v)$	$O(1)$	$O(\deg(u))$
neighbors	$\Theta(N)$	$\Theta(\deg(u))$

ERRANDS Travel from h to $S_1, S_2, \ldots S_k$ subsets of nodes

- → Create k+1 copies of G; G_i
  - $v_i := (v \in G)$'s copy in G_i
  - $\forall v \in S_i$, add edge (v_{i-1}, v_i) w/no weight
- → Run Dijk on h_0 and output shortest path to any $v_k \in G_k$
- → Takes $\Theta(k(|V|+|E|)\log(k|V|))$

MULTIPLE Find shortest path s→t st total weight is a multiple of 5

- → Make 5 copies, G_i $\forall i \in [5]$
  - $\forall$ edges (i,j) in layer $k \in [5]$
    - → Let $m := k + w_{i,j}$ mod 5
    - → connect (i_k, j_m) w/weight w_{ij}
- → Run Dijk. from s → t
- → $\Theta(5(m+n)\lg(5m)) = \Theta((m+n)\log n)$

RED-BLUE Find path through K red blue edges from s → t

- Correctness: We can map any path in G_i so Dijkstra's produces valid shortest following construct
- Create G' w/ $(k+1)^2$ copies of G:
  - → assign each copy tuple $[0,k] \times [0,k]$
  - → $\forall$ e red $= (u,v)$, add edge $(u_{(r,b)}, v_{(r+1,b)})$
  - Run Dijk from $s_{(0,0)} \to t_{(k,k)}$

PMF Let A, B be indep. r.v.s $\in [0,n]$ w/ $Pr[A=k] = a_k$; $Pr[B=k] = b_k$

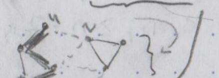
- Compute $Pr[A+B=k]$ given a_i, b_i $\forall i$
- → $Pr[A+B=k] = \sum_{i=0}^{k} Pr[A=i, B=k-i]$
  $= \sum Pr[A=i] \cdot Pr[B=k-i]$

$P_A(x) = \sum_{i=0}^{n-1} a_i x^i$, $P_B(x) = \sum b_i x^i$

$P_A \cdot P_B = \sum_{i=0}^{k} \underbrace{a_i \cdot b_{k-i}} \cdot x^k$

Return coeff of x^k

Given shortest-path labels $d(\cdot)$ for tree T. We remove $(u,v) \in T$ to form S and E(S). Recompute the shortest paths



FACTORIAL Multiply d-bit ints in $O(d^{1.59})$

$a = \prod_{i=0}^{n/2}$, $b = \prod_{i=n/2+1}^{n}$; $c = a \cdot b$

$T(k)$ = time to multiply k ints, each ≤ log n bits

$= 2T(\frac{k}{2}) + O((2 \cdot \log(n))^{1.59})$

↳ Θ: $2^i \cdot (k/2^i \cdot \log n)^{1.59}$

log(n) layers $\le (k \cdot \log n)^{1.59}$

↳ $T(n) = \log(n) \cdot (n \log n)^{1.59}$

$n! \text{ param} = O(n^{1.59} \cdot \log^{2.59}(n))$

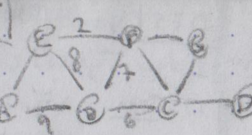
MEDIAN of {1, ..., 101} and pivots

Pivot	Length	k
70	69	51
30	69-30=39	51-30=21
47	22	4
51	0	0

30 47 51 70 101

k = median − pivot

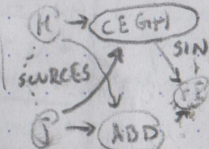
MST



Must be in MST:
- EF: cut AE
- BC: cut B
- CD: only edge to D

Never in MST:
- EB: largest in cycle

A	B	C	D	E	F	G	H	I	J	K
5	5	6	5	6	2	6	6	2	10	10



BATTALIONS We have 2 groups who can never visit same node and go from $s_1 \to t_1$, $s_2 \to t_2$

Construct G' w/ V' = $\{(u_1, u_2) : u_1 \in V, u_2 \in V, u_1 \ne u_2\}$

Edge (u_1, u_2) are adjacent to (v_1, v_2) iff
- $u_1 = v_1$; u_2 adj. to v_2 in G or
- $u_2 = v_2$; u_1 adj to v_1 in G or
- u_1 adj v_1 and u_2 adj to v_2

CHANGE Given $\{c_1, \ldots c_t\} \in [n]$

Find all possible values reachable w/ k st coins: deg nk

$p(x) = \sum_{i=1}^{t} x^{c_i}$ st $(p(x))^k$ contains x^s iff s is the sum of some k coins

SET COVER

Cover at least N/OPT elts

E.g. $\{S_1, \ldots S_m\}$ st $S_i \subseteq \{1 \ldots 1000\}$

→ smallest greedy sol given size = 200 covers

$N = 1000 \therefore 200 \ge \frac{1000}{OPT} \Rightarrow OPT \ge 5$

→ " " given size = 100 covers 10 elts:

$N = 800 \therefore 10 \ge \frac{800}{OPT} \Rightarrow OPT \ge 80$

SPECIAL MST

Given $S \subseteq V$, create an MST s.t. $\forall s \in S$ are leafs in the MST G'

1.) Create an MST of V \ S

2.) $\forall s \in S$: Add lowest-cost edge to $v \notin S$

↳ Impossible: no MST of V \ S or lonely $s \in S$

Alt: Run Kruskals, skipping any edge (s_1, s_2) and any edge connecting an already included $s \in S$

READ THE Q

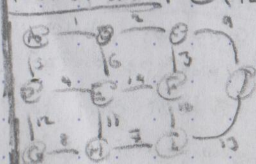
For $u \in S$, $d \in G \setminus S$

$d(u) = \min d(x) + \omega(x,u)$

- → Add u to the PQ w/ priority d(u)
- → Run Dijkstras
- Look at E(S) edges:

$O(|E(S)|) + O(|E(S)| + |S|)\lg(|S|)$

PRIMS/KRU $O(|E|\log(|E|))$ or Kruskals: |V|



{(A,B), (B,C), (C,F), (D,E), (A,D), (H,I)

↳ Not (B,E) ⟺ loop

Pr → Θ H: {(H,I), (H,G), (I,F), (F,G), (C,B), (B,A)}

$T(n) = T(\frac{n}{3}) + T(\frac{2}{3}n) + n$

[→ Guess] $= c \cdot n \lg n$:

$= c(\frac{n}{3})\lg(\frac{n}{3}) + (\frac{2}{3}n)\lg(\frac{2}{3}n) + n$

$\le c(\frac{n}{3})\lg(\frac{2}{3}n) + \ldots$

$= c(n)\lg(\frac{2}{3}n) + n$

$= cn(\log n - \log(\frac{3}{2})) + n$

$\therefore c \ge 1/\log(3/2)$

$T(n) = 3T(n^{1/3}) + \log n$

$x := \log n$

↳ $T(x) = 3T(x/3) + x$

$= \Theta(x \log x)$

$= \Theta(\lg(n) \cdot \lg^2 n)$

$\forall x, y > 1$, $n^x = O(y^n)$

$0 < \frac{\log n}{n} < \frac{\log y}{x+1}$

$(x+1)\log n < n \cdot \log y$

$n^{x+1} < y^n$

$\therefore \frac{n^x}{y^n} < \frac{1}{n}$

$T(n) = 3T(n^{1/3}) + 5n$; $T(1) = 3$

↳ Base @ k: $n^{1/3^k} = 3$

⇒ $k = \log\log(n)$

↳ Total Work: $\sum_{i=0}^{k} 3^i \cdot n^{1/3^i}$

$= n + 3n^{1/3} \ldots 3^{\log_3 n} \cdot n^{1/3^{\log_3 n}}$

$\le (3^{\log_3 n} \cdot n^{1/3}) \cdot \log^2(n)$

$= O(n)$

$= O(S(n))$

$T(n) \le S(n) = 3S(\frac{n}{4}) + 5n$

$= O(n)$

Exchange

1.) Label greedy sol. as $A = \{a_1, \ldots a_k\}$ and opt/arbitrary as $O = \{o_1, \ldots o_m\}$

2.) FSOC: asm A and O differ, i.e.

i. $\exists o \in O, o \notin A \wedge \exists a \in A, a \notin O$

ii. There's 2 conseq. elts $\in A$ that differ in order in O (inversion)

3.) i. Show you can swap an elt in/out

ii. Show " " swap order of elts

And show you can continue swapping so A and O don't differ in poly. time

- → Can't argue $O \ne A$ as there may be many opt. sols
- → Argue why elts exist in A and not O or why 2 elts are out-of-order

max $f_a + 2f_b + 3(f_c + f_d)$

$f_c \le f_b$

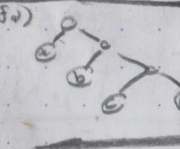
$f_d \le f_b$

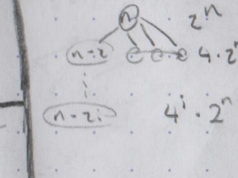
$f_b \le f_a$

$f_c + f_d \le f_a$

$f_a + \ldots f_d = 1$

$f_a \ldots f_d \ge 0$





$4^i \cdot 2^{n-2i}$

$T(n) = T(n-1) + n^2$

$= n^2 + (n-1)^2 + \ldots 1^2$

$= n(n+1)(2n+1)/6$

$= O(n^3)$

$T(n) = T(n-1) + T(n-3) + 1$

$\ge 2T(n-3) + 1$

$= \Omega(2^{n/3})$

$\le 2T(n-1) + 1$

$\le \sum^n 2^i = 2^{n+1} - 1$

$= O(2^n)$

$\frac{d}{dx}\log n = \frac{1}{n}$

$\log(f(n)) = O(\log(g(n)))$

⇒ $f = O(g)$

$f = n^2$, $g = n$

$\lg(f) = \log(n^2) = 2\log(n)$

$= O(\log(g))$

$f = O(g)$

$= O(\log(n))$

$n^2 = O(n)$ ⟹⟸

$T(n) = 4T(n-2) + O(2^n)$

$\sum_{i=0}^{n/2} 4^i \cdot 2^{n-2i}$

$\frac{2^{2i}}{2^{2i}} \cdot 2^n \cdot 2^{-2i} = \sum 2^n$

$= (\frac{n}{2})(2^n) = O(n \cdot 2^n)$

DP | DIGIT DIFF

NICE | $x \in \mathbb{Z}^+$ is 'nice' if any two digits differ by at least 2

13 = nice; 54 = not nice $\in [10^{k-1}, 10^k)$

Find # of nice integers k digits long

$f(d,r)$:= # of nice r-digit ints that start with $d \in [0,9]$

→ Want to find $\sum_{d=1}^{9} f(d,k)$

$$f(d,r) = \begin{cases} 1 & : r = 1 \\ \sum_{d' : |d-d'| \geq 2} f(d', r-1) & : r > 1 \end{cases}$$

→ Calculate $r \in [1,k]$, at each check 9 $= \mathcal{O}(1)$ digits

↳ $\mathcal{O}(k)$ runtime

↳ $\mathcal{O}(1)$ space as bottom-up $f(\cdot, r)$ depends only on $f(\cdot, r-1)$

FLAT SEQ | subseq is 'flat' if

$!7 A_i < A_{i+1} + A_{i-1}$

$$f(i,j) = 2 + \max_{k \in [j,n] : !7 A_j < A_i + A_k} \{ f(j,k) - 1 \}$$

Longest subseq from $A_i \cdots A_n$ st A_i, A_j are first 2 elts

→ $\max(\emptyset) := 0$ catches base-cases, i.e. $f(n-1), f(i,n)$

→ Calculate in $\mathcal{O}(n^3)$

For $i \in \{n-1 \cdots 1\}$:
 For $j \in \{n \cdots i+1\}$:
  $m = 0$
  For $k \in \{j+1 \cdots n\}$:
   $m = \max(m, f(j,k) - 1)$
  $f(i,j) := 2 + m$

rt. $\max_{1 \leq i < j \leq n} f(i,j)$

PHONE | Want to find valid phone # of n-digits, each in L-shape

- Subproblem: $f(n,l)$ is # of possibilities given n remaining digits and @ digit l
- RR: $$f(n,l) = \begin{cases} 1 & : n = 0 \\ \sum_{d \text{ reachable from } l} f(n-1, d) & : \text{ow} \\ 0 & : l \neq 0 \end{cases}$$

LIS | 5 → 4, 1 → 3 | $\mathcal{O}(n^2)$

- Create an edge from $i \to j$ whenever $i < j \wedge a_i < a_j$

↳ Find longest path in DAG

$$L(j) = 1 + \max_{(i,j) \in E} L(i)$$

$\equiv$

For $j \in [n]$: $L(j) = 1 + \max \ldots$

EDIT | S - N O W Y / S U N N - Y } cost = 3

→ Solve for prefix: substitute

$$E(i,j) = \min \begin{cases} \text{diff}(i,j) + E(i-1, j-1) \\ 1 + E(i-1, j) & \text{delete} \\ 1 + E(i, j-1) & \text{insert} \end{cases}$$

$\mathcal{O}(nm)$

KNAP | REP | $\mathcal{O}(nW)$

$K(\omega)$:= max val with bag size ω

$$= \max_{i : \omega_i \leq \omega} (K(\omega - \omega_i) + v_i)$$

ALT: $n(v) = \min_i \{ K(v - v_i) + \omega_i \}$

SINGLE

$K(\omega, j)$:= max val. ... and items $1 \cdots j$

$\mathcal{O}(nW)$ $= \max \{ K(\omega - \omega_j, j-1) + v_j, \; K(\omega, j-1) \}$ (skip)

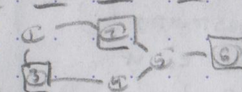
MATRIX MUL.

$C(i,j)$:= min cost of $A_i \cdots A_j$

$$= \min_{k \in [i,j)} \left(C(i,k) + C(k+1,j) + m_{i-1} \cdot m_k \cdot m_j \right)$$

$\mathcal{O}(n^3)$

INDEP. TREE



$\mathcal{O}(|V| + |E|)$

$I(u)$:= size of largest indep. set rooted at u

$$= \max \left\{ 1 + \sum_{w : \text{grandchild}(u)} I(w), \; \sum_{w : \text{child}(u)} I(w) \right\}$$

PALIN SUB-SEQ

$L[i,i] = 1$

$x[i] \neq x[j]$: $\max\{L(i+1,j), L(i,j-1)\}$ — first ≠ last differ

$j = i+1 \wedge x[i] = x[j]$: $L(i,j) = 2$

else: $L(i,j) = L(i+1, j-1) + 2$ — first & last same

CHANGE | ∞ coins | $\mathcal{O}(nV), \mathcal{O}(V)$

Given coins denom. $x_1 \cdots x_n$, make V?

$f(i,V)$:= possible w/ $x_1 \cdots x_i$ to make V?

↳ WTF: $f(n,V)$

- Base: $f(0,0) = T$, $f(0,V) = F$
- RR: $\ldots = f(i-1, V) \;||\; f(i, V - x_i)$

w/ fewest coins: $\mathcal{O}(nV)$

$$f(v) = \begin{cases} 0 & : v = 0 \\ 1 + \min_{j : x_j < v} \{ f(v - x_j) \} & : \exists x_j \leq v \\ \infty & : \text{ow} \end{cases}$$

w/ one coin ea. $\mathcal{O}(nV)$

$$f(i,v) = \begin{cases} 0 & : v = 0 \wedge i = 0 \\ 1 + f(i-1, v - x_i) & : i > 0 \wedge \exists v_i \leq v_i \\ f(i-1, v) & : i > 0 \\ \infty & : \text{ow} \end{cases}$$

MITTENS

- Each month, d_i mittens sold
- Let $D := \sum_i^n d_i$
- Produce m mittens/month
- Produce another pair for cost c
- Storing k left-over costs $h_k \geq 0$
- Let $f(i,r)$ subproblem consider month i w/ left over r → want $f(n,0)$
- RR: $$f(i,r) := \min_{p \in [0, r+d_i]} F(i,r,p)$$

$$F(i,r,p) = h_p + f(i-1, p) + \begin{cases} 0 & : m \geq r + d_i - p \\ c(r + d_i - p - m) & : \text{ow} \end{cases}$$

(p = mittens stored)

→ Base: $\begin{cases} f(0,0) = 0 \\ f(0,+) = \infty \end{cases}$

→ Solve in increasing order of months

→ $n \cdot D$ subproblems, each considering storing $0 - D$ mittens $\Rightarrow \mathcal{O}(nD^2)$

FRIENDS | Given n cities w/ m roads, no cycles

we want to go node $s \to t$ and visit as many friends along the way: $c_1 \cdots c_n$

→ $f(v)$ = max friends visitable from $s \to v$

$f(s) = c_s$, $f(v) = \max_{(u,v) \in E} f(u) + c_v$

$\mathcal{O}(\text{reverse}(m+n) + \sum_{\text{path}} \deg^-(v)) = \mathcal{O}(m+n)$

EXPERT WEATHER

Given n experts (w/ at least one always right)

- If we guess based on the majority who have yet to fail ...

↳ Every fail, $\frac{1}{2}$ experts (at least) who haven't failed prior ⇒ set who have not failed decreases by (at most) $\frac{1}{2}$

- Set starts $1 \cdot 1 = n$, minimally 1, so maximally $\log(n)$ fails

- w/ k experts who never fail...

↳ We go from $n \to 2k-1$ in at most $\lceil \log(n/(2k-1)) \rceil$ fails. Here, majority are always right & never fail again

- w/ one expert who makes at most k fails:

↳ Every fail, pool of those w/ m-fails decreases by at least 1/2.

↳ $m < k$: pool goes from pool size $n \to 0$ before m increases by 1 ∴ w/ every increase, we have at-most $\log(n) + 1$ fails

↳ $m = k$: size $n \to 1$ in $\log(n)$ once

$\Rightarrow k(\log(n) + 1) + \log(n)$

MWU CONVEXITY

Randomized solution is simply a linear combination of deterministic sol., so by convexity it can do no better than deterministic

CARDS | Given m cards, remove min. until one card exist, larger than all others.

$l(i)$ = longest increasing sub-seq end @ exactly i

$$l(i) := \max_{j < i, c_j < c_i} \{ 1 + l(j) \} \;|\; l(0) = 1, \; l(j) = 1 \text{ if } c_j \geq c_i$$

$r(i)$ = longest inc. SS starting @ exactly i

$$r(i) := \max_{j > i, c_j < c_i} \{ 1 + r(j) \} \;|\; r(n-1) = 1$$

rt: $m + 1 - \max_i \{ l(i) + r(i) \}$

2m sub-prob. each taking m time $\Rightarrow \mathcal{O}(m^2)$

w/ cycles: decompose into SSCs in $\mathcal{O}(n+m)$ and run prior algo

Zero-Sum Games

Row : max. actions, n - Primal
Col = min. actions, m - Dual

$$M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,m} \\ \vdots & & \vdots \\ M_{n,1} & & M_{n,m} \end{bmatrix}$$

Row: $\max_x \min_y x^T M y \equiv$ Col: $\min_y \max_x y^T M x$

$\max z : \forall j \in [m] : \sum_{i=1}^{n} M_{i,j} \cdot p_i \geq z$

$\sum_{i=1}^{n} p_i = 1, \ \forall i \in [n] : p_i \geq 0$

$\min \omega : \forall i \in [n] : \sum_{j=1}^{m} M_{i,j} \, q_j \leq \omega$

$\sum_{j=1}^{m} q_j = 1, \ \forall j \in [m] : q_j \geq 0$

By strong duality, opt value is same if min/max probabilities are fixed first

$\max_x \min_y \sum_{ij} M_{ij} x_i y_j$ } row first
$\equiv$
$\min_y \max_x \sum_{ij}$ ↓ } col first

$\min_{c \in C_{pure}} \max_{r \in R_{pure}} (\ldots) \geq \max_{r \in R_{pure}} \min_{c \in C_{pure}} (\ldots)$

$\min_{c \in C_{pure}} \max_{r \in R_{mix}} (\ldots) \geq \min_{c \in C_{mix}} \max_{r \in R_{mix}} (\ldots)$

$\min_{c \in C_{pure}} \max_{r \in R_{mix}} (\ldots) = \min_{c \in C_{pure}} \max_{r \in R_{pure}} (\ldots)$

OPT Row strat: $\begin{bmatrix} a & -b \\ -c & d \end{bmatrix}$

$\max_z : \left.\begin{matrix} z \leq a x_1 + c x_2 \\ z \leq -b x_1 + d x_2 \end{matrix}\right\}$ Equal $\Rightarrow (a+b) x_1 = (c+d) x_2$

$x_1 = \frac{c+d}{a+b+c+d}, \ x_2 = \frac{a+b}{a+b+c+d}$

$\hookrightarrow z = a x_1 - c x_2 = \frac{ad-bc}{a+b+c+d}$

OPT Col strat:

$\min_z : \left.\begin{matrix} z \geq a y_1 - b y_2 \\ z \geq -c y_1 + d y_2 \end{matrix}\right\} \Rightarrow (a+c) y_1 = (b+d) y_2$

$y_1 = b+d/(a+\ldots+d)$

$y_2 = a+c/(\ldots)$

$\therefore z = a y_1 - b y_2$

$= \frac{ad-bc}{a+b+c+d}$ [Weak duality] primal (row) at least dual (col)

MWU

Offline OPT ≡ total loss of best ≡ loss had picked best

- $x_i^{(t)}$: prop. trust expert i on day t
- $l_i^{(t)}$: loss on t if invest all in i
- Total Regret: $R_T = \sum_{t=1}^{T} \sum_{i=1}^{n} x_i^{(t)} l_i^{(t)}$ } our loss $- \underbrace{\min_{i \in [n]} \sum^{T} l_i^{(t)}}$

Update as...
- $\omega_i^{(0)} = 1$
- $\omega_i^{(t)} = \omega_i^{(t-1)} \cdot (1-\varepsilon)^{l_i^{(t-1)}} \leq$ (best exp. loss)
- $x_i^{(t)} = \omega_i^{(t)} / \sum_{i=1}^{n} \omega_i^{(t)}$ $\left(\leq (1 - \varepsilon l_i^{(t-1)})\right)$

If $\varepsilon \in (0, 1/2]$, $l_i^{(t)} \in [0,1]$

Loss alg incurs $\{ L - $ OPT (off opt.) $\Rightarrow R_T \leq \varepsilon T + \frac{\ln(n)}{\varepsilon}$

t [time steps] $\propto$ runtime
i [experts] $\propto$ memory

min loss alg

Say best exp. costs 20, $t \in [100]$, $\varepsilon = 1/2$

$R_T =$ [our loss] $-$ [best exp. loss]

$=$ loss $- 20 \leq (\frac{1}{2})(100) + \ln(2)/_{1/2}$

$\Rightarrow$ loss $\leq 70 + 2\ln(2) \approx 71.39$

Say $t \in [25]$, set ε to min R_T

$\frac{d}{d\varepsilon} R_T \leq \frac{d}{d\varepsilon} \left(\varepsilon T + \ln(n)/_{\varepsilon} \right)$

$0 \leq T - \ln(n)/\varepsilon^2$

$\varepsilon = \sqrt{\ln(n)/T} = \sqrt{\ln(2)/25} \approx 0.17$

Say $t \in [140]$, $\underbrace{\text{exp. \#1}}_{\text{never wrong}}$ costs 0, $\underbrace{\text{\#2}}_{\text{always wrong}}$ costs 1

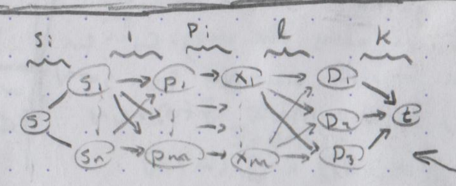
$\varepsilon = 0.01$

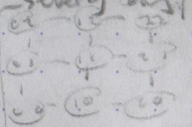
$\Rightarrow \omega_1 = 1 \qquad \Rightarrow \omega_2 = (1-\varepsilon)^{140}$
$= (.99)^{140} \approx 1/4$

$\therefore p_1^{(141)} = \frac{\omega_1}{\omega_1 + \omega_2} = \frac{1}{1+1/4} = \frac{4}{5}$

If $\omega_i^t > \omega_j^t$ ...
- Expert i had smaller total loss
- MWU more likely to pick i @ $t+1$

But we cannot say that i is less likely to experience loss less at $t+1$, as this can be arbitrary / indep. of first t days





→ every day

What is shortest ST path, given t days of history?

→ Experts map to paths (6) =: n

→ Say $l_i^{(t)} \in [0, 4]$ and $R_T \leq 8\sqrt{T \ln(n)}$

to bound this to $T/1000$ st...

$T/10^3 \leq 8\sqrt{T \ln(6)} \Rightarrow T > (8 \cdot 10^3)^2 \cdot \ln(6)$

If we have $l_i \in [a, b]$, we can normalize to $[0,1]$ as $l_i^{(t)} := \frac{l_i^{(t)} - a}{b-a}$ and bound Regret as...

$R_T \leq 2 \cdot (b-a)\sqrt{T \cdot \ln(n)}$

FLOW

$\underbrace{\text{MAX-FLOW}}_{\text{Primal}} : \underbrace{\text{MIN-CUT}}_{\text{Dual}}$

$\hookrightarrow \max \text{size}(f) \leq \min_{\text{cut}} \text{cap}(L, R)$

- FF (DFS): $O(f \cdot E)$
- EK (BFS): $O(V \cdot E^2)$ or $O(V^2 \cdot E)$

GYM

Given D days and n instructors w/range $\{[a_1, b_1] \ldots [a_n, b_n]\}$ we want exactly K instructors every day

Sol 1: Create s, t ; $i \in [n]$

If $a_j = b_i + 1$, connect $i \to j$

If $a_i = 0$: connect $S \to i$

" $b_i = D$: " $i \to t$

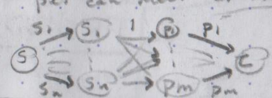
Sol 2: Create edge $u \to v$, for $u, v \in [0 \ldots D]$ w/capacity of # instructors available from $v+1$ to u

↳ Say s maps to 0 index

↳ Edge $D \to t$ w/capacity K

PETS

- i^{th} student wants to meet exactly s_i pets
- j^{th} pet can meet at most p_j students



- Meeting schedule over 3 days
- max of K meetings/day
- pet meet at most l/day

LP

$\max_{c^T x} Ax \leq b \qquad \min_{b^T y} A^T y \geq c$

- x is set of variables we operate on
- c is the productivity of each in x
- A corresponds to constraints on x
- b holds max values for each constraint

- $\max c^T x \equiv \min(-c^T x)$
- $x_1 \leq b_1 \equiv x_1 + s_1 = b_1, \ s_1 \geq 0$
- $x_2 \geq b_2 \equiv -x_2 + s_2 = -b_2, \ s_2 \geq 0$
- $b_2 \leq x_3 \leq b_1 \equiv x_3 \leq b_1 \wedge x_3 \geq b_2 \equiv \ldots$
- $x_1 + x_2 + x_3 \leq b_3 \equiv x_1 + x_2 + x_3 + s_4 = b_3, \ s_4 \geq 0$
- $\min_{y_1, y_2} \max \equiv \min_t : y_1 \leq t, \ y_2 \leq t$
- $x_4 \in \mathbb{R} \equiv x_4 = x^+ - x^-, \ x^+, x^- \geq 0$

Given n variables, m inequalities...

→ n ineq. hold at every vertex
→ $n-1$ ineq. are same between vertex neighbors

At least

Primal → Dual

OPT $\Rightarrow x_1 = x_2$

$\max 4x + 7y$

$\left.\begin{matrix} x + 2y \leq 10 \\ 3x + y \leq 14 \\ 2x + 3y \leq 11 \end{matrix}\right]$ $\begin{matrix} a(\ldots) \leq 10a \\ b(\ldots) \leq 14b \\ c(\ldots) \leq 11c \end{matrix}$

$x, y \geq 0$

$x(a + 3b + 2c) + \ldots \geq 4$
$\equiv y(2a + b + 3c) \ldots \geq 7$
$\leq 10a + 14b + 11c$

$\Rightarrow \min 10a + 14b + 11c$
$a + 3b + 2c \geq 4$
$2a + b + 3c \geq 7$
$a, b, c \geq 0$

$\min_{y^T b} = \max_{c^T x}$
$= c^T(A^{-1} b)$
$\therefore y^T = c^T A^{-1}$

ABS

$\min \sum^{n} \underbrace{|y_i - (a + b x_i)|}_{=: z} \to \begin{cases} y_i - (a + b x_i) \leq z_i \\ -y_i + (a + b x_i) \leq z_i \end{cases} \forall i \in [n]$

Weak Duality] If $x \in \mathbb{R}^n$ feasible for P } $A\vec{x} \leq \vec{b}$
$y \in \mathbb{R}^m$ " " D } $A^T y \geq \vec{c}$

$\Rightarrow c^T x \leq (A^T y)^T x = y^T (A x) \leq y^T b = b^T y$

$\therefore P$ unbound $\Rightarrow D$ necessarily infeasible,
D " " P " "

Strong] If either P or D has a finite opt then so does the other, these opts exist + coincide

Randomized Algos

Chebyshev: $\mathbb{P}[|X - \mathbb{E}(X)| > \lambda] < \frac{Var(X)}{\lambda}$

$\mathbb{E}(X) = \sum_{k=-\infty}^{\infty} k \cdot \mathbb{P}(X=k)$ | Markov: $\mathbb{P}[X > \lambda] < \frac{\mathbb{E}(X)}{\lambda}$

Karger | Find min-cut by 'contracting' random edges

-> Fails [when all min-cut edge(s) are gone

$\frac{n}{n+1} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n-1} \cdots \frac{2}{3} = \frac{2}{n+1}$

Hashing

$m :=$ # hash bins
$n :=$ # DB elts.

$\mathbb{E}(\text{Time to query } x) \leq \mathbb{E}(\underbrace{T_h}_{\text{Hash Func.}} + C \cdot (\text{size of } x\text{'s LL}))$

Linearity of expect. $\longrightarrow$ $= T_h + C \cdot \mathbb{E}(\sum_{i=1}^{n} Z_i)$

Z_i: Indicator for if $h(j_i) = h(x)$ $\longrightarrow$ $= T_h + C \cdot \sum \mathbb{E}(Z_i)$

w/ $j_1 \cdots j_n$ DB keys $= T_h + C \cdot \sum \mathbb{P}(h(j_i) = h(x))$

$= T_h + C \sum_{i=1}^{n} \sum_{t=1}^{m} \mathbb{P}(h(j_i) = t \wedge \underbrace{h(x) = t}_{})$

$h(\cdot)$ is random, $\mathbb{E}(Z_i)$ is $= \frac{1}{m}$ $\longrightarrow$ $= T_h + C(n \cdot m \cdot \frac{1}{m^2})$

$= T_h + C(\frac{n}{m})$

K-indep | Set of funcs mapping $[U] \mapsto [m]$ is " iff:

$(\forall x_1 \neq x_2 \neq \ldots)(\forall y_i)\left(\mathbb{P}((h(x_1) = y_1) \wedge \ldots \wedge (h(x_k) = y_k)) = \frac{1}{m^k}\right)$

-> Each $h \in \mathcal{H}$ is deterministic, but sampled from $\mathcal{H}$ uniform-randomly

- $\mathcal{H}$ is universal if.. $\forall k \in U, h \in \mathcal{H}$: $\mathbb{E}[\text{colls of } h(k)] = \underbrace{|U|-1}_{|U \setminus \{k\}|} / \underbrace{m}_{\text{Bins}}$

$\underbrace{(\forall x_1 \neq x_2)(\mathbb{P}(h(x_1) = h(x_2))}_{\text{chance of collision}} \leq 1/m$

↳ 2-wise indep $\Longrightarrow$ universal

-> Specifying an $h \in \mathcal{H}$ takes bits $\sim \mathcal{O}(\log_2 |\mathcal{H}|)$

↳ **seeding**

$\mathcal{O}(\log m^U)$

$\mathcal{O}(U \log(m))$

↳ If we limit $U = m = p$ (prime #)

and say $\mathcal{H} = \{ax + b \,\%\, p \mid a, b \in [p]\}$

we have $|\mathcal{H}| = |a \times b| = p^2$ $\therefore \log|\mathcal{H}|$

while still 2-wise indep $= \mathcal{O}(\log p)$

1) ~~Las~~ Las Vegas: Always correct, runtime is RV (i.e Quicksort)

$(\forall x)(\mathbb{E}_r(\text{runtime } A(x,r)) \leq \varepsilon)$

2) Monte Carlo: Fixed runtime, correctness is RV

$(\forall x)(\mathbb{P}_r(A(x,r) \text{ is correct}) \geq 1 - \varepsilon)$

LV → MC (ZPP → RP) | Let $A \in ZPP$ st $A = \mathcal{O}(n^k)$

Construct A' as: run A for $2n^k$ steps

$\begin{cases} \text{if } A \text{ ever terminates, return its output} \\ \text{else: rt 'No'} \end{cases}$ Error

- we want to bound the chance we ~~lazily~~ falsely rt 'No' that is, that we terminate in $2n^k$
- Let X be rv for # steps. We know $\mathbb{E}[X] \leq n^k$
- Markov's gives us: $\mathbb{P}[X \geq 2n^k] < n^k / 2n^k = 1/2$

i.e. we give correct answer w/ $p = 1/2 \Rightarrow A' \in RP$

Perfect Hashing | h s.t. $\forall k_1 \neq k_2$ then $h(k_1) \neq h(k_2)$

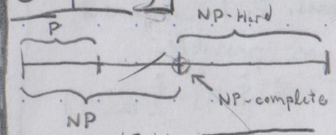
-> Contrapositive bday paradox: if few people are in a room + many calendar days then likely to not share bday

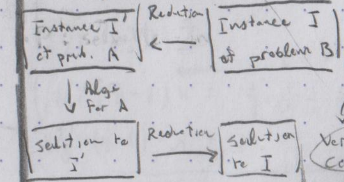
-> $\mathbb{P}(\exists \text{ collision}) = \mathbb{P}(\# \text{collisions} \geq 1) = \frac{\mathbb{E}(\# \text{ colls})}{1}$

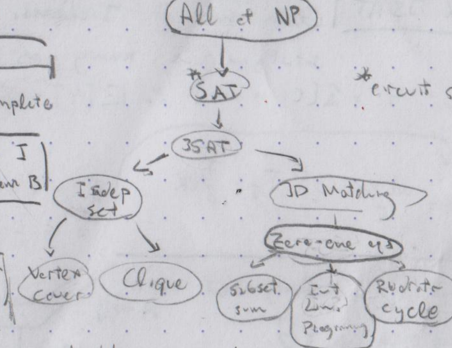
- $Z_{ab} = \frac{1}{m}$ by 2 wise-indep. $= \mathbb{E}(\sum_{a<b} Z_{ab})$
- $\binom{n}{2}$ elts gives $\ldots = \binom{n}{2}/m$, $= \sum \mathbb{E}(\text{ " })$

$\ldots < \frac{1}{2}$ when $m = n^2$ $= \sum_{a<b} \mathbb{P}(h(j_a) = h(j_b))$

Complexity







- P: Decision problem solvable in polynomial

-> Also in NP: no subroutine req., just use algo to solve

- NP: Can be verified in polynomial time

-> solution is poly. $\Rightarrow$ solvable in poly. space

- NP-Complete: All NP can reduce to

-> To prove: show (1) NP-Hard (2) in NP

can reduce to some A ... and A can be verified in poly.

- NP-Hard: Any problem at least as hard as an NP-complete

-> if solvable in poly $\Longrightarrow$ all NP solvable in poly. (P = NP)

Reductions | A reduces to B implies..

$\exists$ sol I' to A $\Longrightarrow$ $\exists$ sol I to B $\Longrightarrow$ $\exists$ sol I' to A

$\exists$ poly time algo to convert A's inputs to B, so we can use B to solve A

so B is at least as hard as A:

Contrap.

- $B \in P \Rightarrow A \in P$, $A \notin NP$
- $B \in NP \Rightarrow A \in NP$, $A \in P \cup NP$

P. Easy	Hard	Easy	Hard
2SAT, HORNSAT	3SAT	Linear Prog.	Integer LP
MST	TSP	Euler Path	Rudrata path
Shortest Path	Longest P	Min. cut	Balanced cut
Bipartite Matching	3D match.		
Unary knapsack	Knapsack		
IS on trees	IS		

MAX 3SAT | Given ~~sets~~ collection of size k 3-clauses. Find assignment satisfying as many as possible

Let $Y_i := \mathbb{1}[C_i \text{ is sat.}]$ $\therefore \mathbb{E}[Y] = \mathbb{E}[\sum_{i=1}^{k} Y_i] = \sum_{i=1}^{k} \mathbb{E}[Y_i]$

$\mathbb{P}[Y_i = 0] = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$

$= \sum^{k} (\mathbb{P}[Y_i = 1] \cdot 1)$

$= \sum (1 - \mathbb{P}[Y_i \neq 0])$

$= k \cdot (1 - \frac{1}{8}) = \frac{7}{8} k$

$\therefore \min_I OPT_I = \frac{7}{8} k$

H = univ. hashing set over $A \to B$

$a, b \in A$, $a \neq b$

$\# h \in \mathcal{H}$ st $h(a) = h(b) < \frac{|\mathcal{H}|}{|B|}$

A is correct 1/2 time

B runs A indep. T times, re orb. output if any call gives output. else 'FAIL'

→ How big to make T s.t. B gives 'FAIL' w/ $\delta = 0.05$

$(\frac{1}{2})^T \leq 0.05 \Rightarrow \frac{1}{(\frac{1}{2})^T} \geq \frac{1}{0.05}$

$\log_a(a^b) = b$

$\log_2(2^T) \geq \log_2(\frac{1}{0.05})$

$T \geq \log_2 20$

$\therefore T \geq 5$

IS APPX

while G ≠ empty:
- v = randNode(G)
- I.append(v)
- G.rmNodes(v, v.neighbors())

If all $v \in G$ have degree $\leq d$ then each iter we increase $|I|$ by 1 and rm $O(d+1)$ nodes $\therefore \Omega(|V|/(d+1))$ iterations

→ Since $K := |OPT_G| \leq |V|$

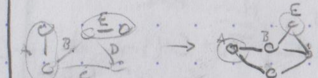
$|I| \geq \frac{|V|}{d+1} \geq \frac{k}{d+1}$ } Factor

Matching: Given $\{G, k\}$, want $\{e_1 \dots e_k\}$ st no pairs share a node

Indep. set: Given $\{G, k\}$, want $\{v_1 \dots v_k\}$ st no pairs share an edge (adj.)

→ II is believed to not have a poly. reduction to [Bipartite Matching] because IS ⊂ NP-comple, II ⊂ P so this'd imply P = NP

→ Matching reduces to IS as for each edge, create v'; $\forall (e_1, e_2)$ sharing a common node; add edge $e'_{12} = (v'_1, v'_2)$



Far-Away Pts: Given $\{d[i][j]\}_{\forall i,j \in [n]}^{n\{K,D\}}$ that satisfy triangle ineq., want K nodes all at least distance D from one another

IS → IIS: Run Floyd-Warshall in $O(|V|^3)$ to construct $d[i][j]$; pass this & K = s, D = 2.

FarPts sol: No nodes connected by an edge ⇒ no common neighbors

IS sol: No adj ⇒ $D[i,j] \geq 2$

Vertex Cover: Set of nodes st. at least one endpoint of all $e \in E$ is included

→ Compliment of IS

→ Greedy 2-apprx algo

$x = \min_i \max_j A_{ij}$

$y = \max_j \min_i A_{ij}$

$\Rightarrow x \geq y$

UNIQUE STRAT

same col entries $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ ⟺

col: not unique

row: unique

Mix → Pure: change $x_i \geq 0$ to $x_i \in \{0, 1\}$

UNION | Rank := Height w/no relaxations

1. Rank strictly increases, following parent ptrs
2. ↳ $(\forall x)(\text{rank}(x) < \text{rank}(\pi(x)))$
2. Any root x of rank k has $\Omega(2^k)$ descendants
3. There's $\Omega(n/2^k)$ nodes w/rank K

union: if equal ranks: Goto represent of each, break arb. and increase new root rank by 1

else: lesser rank's head → higher rank