$$\sum_{i}^{n} i = \frac{n(n+1)}{2}$$

$$\sum_{i=0}^{n} \alpha^i = \frac{\alpha^{n+1}-1}{\alpha-1}$$

$$S = 1 + \alpha + \alpha^2 + \dots + \alpha^n$$
$$\alpha S = \alpha + \alpha^2 + \dots + \alpha^n + \alpha^{n+1}$$
$$\alpha S - S = \alpha^{n+1} - 1$$
$$S(\alpha - 1) = \alpha^{n+1} - 1$$
$$S = \frac{\alpha^{n+1}-1}{\alpha-1}$$

when $\alpha < 1$ $\quad \sum_{0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$

$$\sum_{0}^{n} i\alpha^i$$

$$\alpha + 2\alpha^2 + 3\alpha^3 + \dots + n\alpha^n$$

$$\alpha + \alpha^2 + \alpha^3 + \dots + \alpha^n$$
$$\alpha^2 + \alpha^3 + \dots + \alpha^n$$
$$\alpha^3 + \dots + \alpha^n$$

$$\sum_{0}^{n} i\alpha^{i-1} \quad \frac{(n+1)\alpha^{n+1}}{\alpha-1} - \frac{\alpha^{n+1}-1}{(\alpha-1)^2}$$

Harmonic Series

$$H_n = \sum_{i}^{n} \frac{1}{i}$$

$H_1 = 1$

$H_2 = 1 + \frac{1}{2}$

$H_3 = 1 + \frac{1}{2} + \frac{1}{3}$

$H_n = \underbrace{1}_{1} + \underbrace{\frac{1}{2}}_{\frac{1}{2}} + \underbrace{\frac{1}{3} + \frac{1}{4}}_{\frac{1}{2}} + \underbrace{\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}}_{\frac{1}{2}} + \frac{1}{9} + \dots + \frac{1}{16} + \dots + \frac{1}{n}$

$\log n$ groups, each adds up to < 1

$H_n < \log_2 n + 1 \quad H_n = O(\log N)$

$\ln(n+1) \le H_n \le \ln n + 1$

Recurrence Relations

$$T(n) = \begin{cases} d, & n=1 \\ ST(\frac{n}{k}) + cn^b, & n>1 \end{cases}$$

let $n = k^p$

$T(n) = ST(\frac{n}{k}) + cn^b$

$= S(ST(\frac{n}{k^2}) + c(\frac{n}{k})^b) + cn^b$

$= S^2T(\frac{n}{k^2}) + cS(\frac{n}{k})^b + cn^b$

$\vdots$

$= S^iT(\frac{n}{k^i}) + cS^{i-1}(\frac{n}{k^{i-1}})^b + cS^{i-2}(\frac{n}{k^{i-2}})^b + \dots + cS(\frac{n}{k})^b + cn^b$

$$T(n) = S^pT(\frac{n}{k^p}) + cn^b \sum_{0}^{p-1} (\frac{s}{k^b})^d$$
$$= dS^p + cn^b \sum_{0}^{p-1} (\frac{s}{k^b})^d$$

$\frac{S}{k^b} < 1 \to O$

$\frac{S}{k^b} = 1 \to P$

$\frac{S}{k^b} > 1 \to \infty$

Case 1: $S = k^b$

$T(n) = dk^{bp} + cn^b p$

$= dn^b + cn^b \log_k n$

$= O(n^b \log n)$

* $n = k^p$
$p = \log_k n$

When $S \neq k^b$

$$\sum_{0}^{p-1} (\frac{s}{k^b})^i = \frac{1-(\frac{s}{k^b})^p}{1-\frac{s}{k^b}}$$
$$= \frac{1-\frac{n^{\log_k S}}{n^b}}{1-\frac{s}{k^b}}$$

$$T(n) = dn^b + \frac{cn^b - cn^{\log_k S}}{1-S/k^b}$$

Case 2: $S < k^b$

$T(n) = O(n^b)$

Case 3: $S > k^b$

$T(n) = O(n^{\log_k S})$

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

$a=2$ $\quad$ $s = k^b$
$k=2$ $\quad$ $2 = 2^1$
$b=1$ $\quad$ $O(n \log n)$

$T(n) = 3T\left(\frac{n}{2}\right) + n$ $\quad$ $a=3$, $k=2$, $b=1$

$O(n^{\log_2 3})$

$T(n) = 3T\left(\frac{n}{2}\right) + 1$ $\quad$ $a=3$, $k=2$, $b=0$

$O(n^{\log_2 3})$

$T(n) = 2T\left(\frac{n}{2}\right) + n \log n$

$O(n \log^2 n)$

$T(n) = T(\sqrt{n}) + 1$

$O(\log \log n)$

$T(n) = 2T(\sqrt{n}) + 1$

$\Rightarrow 2T e^{\frac{1}{2}\log n} + 1$ $\quad$ sub $x = \log n$ then do masters theo

$\Rightarrow 2T\frac{x}{2} + 1$

$O(x) = O(\log n)$

$T(n) = \sqrt{n}\, T(\sqrt{n}) + 1$

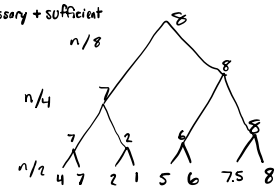
Sorting

Compare-exchange

Operation: C(A, i, j)

1) Find max element of A
   ↳ Linear scan, n-1 comparisons necessary + sufficient

2) Find max and 2nd max element of A
   ↳ 2 scans → 2n-3 comparisons



comp: N-1+log N

Lower bound on # comparisons
To find median of n elements $\geq 3\left\lfloor \frac{N}{2} \right\rfloor$

Adversary: T, B, M (3 sets)

T, B ← ∅ lo = 1
M ← {x₁, ..., xₙ} hi = n

Alg compares $x_i > x_j$?

Case 1: $x_i \in M, x_j \in M$ → Yes
- $x_i \leftarrow$ hi, hi ← hi − 1
- $x_j \leftarrow$ lo, lo ← lo + 1
- T ← T ∪ {x_j}
- B ← B ∪ {x_i}

Case 2: $x_i, x_j \in$ (T or B) → gives correct answer

Case 3: $x_i \in T, x_j \in M$
YES

Case 4: $x_i \in B, x_j \in M$
No

$\geq \left\lfloor \frac{N}{2} \right\rfloor$ queries to commit all values

$\geq 2\left[\left\lfloor \frac{N}{2} \right\rfloor + 1 - 1\right]$ to confirm median

$\geq 3\left\lfloor \frac{N}{2} \right\rfloor$ comparisons

σ: 4 7 9 8 6 3 1 2 5
1: 4 7 9 8 3 6 1 2 5
2: 4 7 3 8 1 6 2 6 5
3: 4 3 7 1 8 2 6 5 6
4: 3 4 1 7 2 8 5 6 6
5: 3 1 4 2 7 5 8 6 9
6: 1 3 2 4 5 7 6 8 9
7: 1 2 3 4 5 6 7 8 9

Oblivious compare/exchange alg: C/E operations are fixed in advance (for all input values)

0/1 principle: An oblivious C/E algorithm sorts every input sequence IFF it sorts every 0/1 sequence

input: $x_1, x_2, \ldots, x_n$

Sorted: $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}$ π: permutation of 1, ..., n

Suppose that c/e alg does not sort every sequence

Algorithm: $x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}$ $x_{\pi(k)} = x_{\sigma(k)}$

correct sequence: $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)}, \ldots, x_{\pi(n)}$

algo answer: $x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)}, \ldots, x_{\sigma(n)}$

Let k be the smallest index: $x_{\pi(k)} < x_{\sigma(k)}$

$x_i^* = \begin{cases} 0, & x_i \leq x_{\pi(k)} \\ 1, & x_i > x_{\pi(k)} \end{cases}$

Lemma:
Input: $x_1, \ldots x_n$: output $y_1 \ldots y_n$ (sorted ↗)
Input: $f(x_1) \ldots f(x_n)$: output $f(y_1) \ldots f(y_n)$
f: monotonically non-decreasing

9/14

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

$\frac{n}{2}(\log n - 1) < \log n! < n \log n$

$\log n! = \Theta(n \log n)$

$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \ldots + \frac{x^k}{k!} + \ldots = \sum_{i=0}^{\infty} \frac{x^i}{i!}$

$|x| \le 1$:

$1 + x + x^2 > e^x > 1 + x$

Sterling's Approx.

$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$

$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \le n! \le \sqrt{2\pi n}\left(\frac{n}{e}\right)^{n + \frac{1}{6n}}$

binomial coeffs: $\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!} = \frac{n!}{k!(n-k)!}$

$< \frac{n^k}{k!} < \left(\frac{ne^k}{k^k}\right)$ $\qquad \binom{n}{n/2} < \left(\frac{2ne}{n}\right)^{n/2}$

$e^k > \frac{k^k}{k!}$

$\frac{1}{k!} < \frac{e^k}{k^k}$

sorting algorithms: min $n!$ leaves

Element Distinctness

Input: Array A $x_1, \ldots, x_n$

OUTPUT: Yes IFF $\forall i \neq j$, $x_i \neq x_j$

Suppose that arrays $X = [x_1, \ldots, x_N]$ and $Y = [y_1, \ldots, y_n]$ give identical answers for every comparison (they follow the same path from root→leaf)

Let $0 \le \lambda \le 1$. consider array $Z = [z_1, \ldots, z_n]$ where $z_i = \lambda x_i + (1-\lambda) y_i$

Claim: Z follows same path as X, Y.

$x_i > x_j \iff y_i > y_j$

$\lambda(x_i) + (1-\lambda)y_i > \lambda(x_j) + (1-\lambda)y_j$

$S_\pi = \{X : x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)}\}$

$S_{\pi'} = \{X : x_{\pi'(1)} < x_{\pi'(2)} < \ldots < x_{\pi'(n)}\}$

$A \in S_\pi$, $B \in S_{\pi'}$

s.t. A contains distinct elems, as does B

Suppose A and B end up in the same YES leaf

A: $x_i > x_j$

B: $y_j > y_i$

$z_i = \lambda x_i + (1-\lambda) y_i \quad = \quad z_j = \lambda x_j + (1-\lambda) y_j$

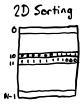
$\lambda = \frac{y_i - y_j}{(x_j - x_i) + (y_i - y_j)}$

In n steps, with n 0's and 1's, prove it will be sorted

assume k 0's and $N-k$ 1's

think about leftmost/rightmost 0 and 1

2D Sorting



row: mixed if contains both 0's and 1's

pure, o.w.

Row sort + Column Sort: for each pair of consecutive rows

get pure row produced

Initially: $\le N$ mixed rows

After 1 round $\le N/2$ mixed rows

2 rounds $\le N/4$ mixed rows

SORT: $N \times N$ grid.

1: REC sort each $\frac{N}{2} \times \frac{N}{2}$ quadrant

2: REVERSE EVERY OTHER ROW

3: COLUMN SORT ($N \times N$)

4: TWO ROWS OF ROW+COL SORT ($2 \times N$)

$T(N) = T\left(\frac{N}{2}\right) + \frac{N}{2} + SN$

$T(N) = T\left(\frac{N}{2}\right) + cN$

1/0 Principle Proof

Ex1:

10000111
01000111
00100111
00010111
00001111

Ex2:

5 11110000
1 11101000
2 11010100
3 10101010
4 01010101
5 00101011
6 00010111
7 00001111

* in worst case
- rightmost 1 takes 5/1 turns to get in position
- the position for the next 1 is shifted by one but it will take 1 iteration for it to start moving, so n/2+1
- The last 1 needs n/2-1 iterations to start moving, so it will take n-1 iterations to get in place

Ex3:

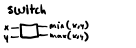
5 10101010
01010101
00101011
00010111
00001111

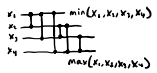
Ex4

11001100
10101010
01010101
00101011
00010111
00001111

9/19

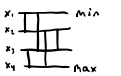
switch





SIZE(network) = #switches : 6
DEPTH(network): 5

SIZE(SORTING NETWORK) = $\Omega(N \log N)$
DEPTH(SORTING NETWORK) = $\Omega(\log N)$



SIZE=5
DEPTH=3

merge sort idea

$S(N) \leq 2S(N/2) + M(N)$

$D_S(N) \leq D_S(N/2) + D_M(N)$

$S(2) = 1$

$D_S(2) = 1$

SIZE=
DEPTH=O(N)

not efficient

Odd-even merge

A: 2 3 4 8
B: 1 5 6 7

even(A): 2 4 odd(A): 3 8

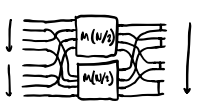
odd(B): 5 7 even(B): 1 6

Merge (A, B): A, B sorted
1. $L_1 \leftarrow$ merge (even(A), odd(B))
2. $L_2 \leftarrow$ merge (odd(A), even(B))
3. $L \leftarrow$ Shuffle (L_1, L_2)
4. C/E consecutive pairs

2 4 5 7 1 3 6 8

2 1 4 3 5 6 7 8

1 2 3 4 5 6 7 8



$D_M(N) \leq D_M(N/2) + 1$ $D_M(N) = \log N$

$D_S(N) \leq D_S(N/2) + \log N$ $D_S(N) = O(\log^2 N)$

$M(N) \leq 2M(N/2) + N/2$ $M(N) = O(N \log N)$

$S(N) \leq 2S(N/2) + O(N \log N)$ $S(N) = O(N \log^2 N)$

Proof using 0/1 Principle

A: $\overbrace{0\ldots0}^{a}\ \overbrace{1\ldots1}^{n-a}$

B: $\overbrace{0\ldots0}^{b}\ \overbrace{1\ldots1}^{N-b}$

L_1: $O_1 = \lceil \frac{a}{2} \rceil + \lfloor \frac{b}{2} \rfloor = c$

L_2: $O_2 = \lfloor \frac{a}{2} \rfloor + \lceil \frac{b}{2} \rceil = d$

$c=d$ $c=d+1$ $\rightarrow \overbrace{0\ldots0}^{c+d}\ 1\ldots1$

$c=d-1 \rightarrow 0\ldots01\ 01\ldots1$

Parallelize seemingly sequential programs

add bits

C_{i+1}

$a_n \cdots a_4\, a_3\, a_2\, a_1$

$b_n \cdots b_4\, b_3\, b_2\, b_1$

$\vee$: +

$\wedge$: ·

$a \vee b$: $a+b+ab$

$a \oplus b$: $a+b$

$s_i = a_i \oplus b_i \oplus c_{i-1}$

$c_i = (a_i \wedge b_i) \vee ((a_i \oplus b_i) \wedge c_{i-1})$

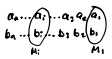
$c_i = a_i b_i + (a_i + b_i) c_{i-1}$

$$\begin{pmatrix} c_i \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} a_i + b_i & a_i b_i \\ 0 & 1 \end{pmatrix}}_{M_i} \begin{pmatrix} c_{i-1} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} c_i \\ 1 \end{pmatrix} = M_i \begin{pmatrix} c_{i-1} \\ 1 \end{pmatrix}$$

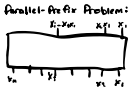
$$= M_i M_{i-1} \begin{pmatrix} c_{i-2} \\ 1 \end{pmatrix}$$

$$= M_i M_{i-1} M_{i-2} \begin{pmatrix} c_{i-3} \\ 1 \end{pmatrix} = M_i \cdots M_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



computing the parallel-prefix problem

Parallel-Prefix Problem:



9|21

Randomness

3 card monty:

Any cost:

$\frac{1}{3}(1+2+3) - 2 = 0$

1. Deterministic — Worst-case
2. Average Cost of det alg — Have to know distribution
3. Random Guess — Every strategy of adversary

Random Guess:

E(gain):

$\frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 0 + \frac{1}{3} \cdot -1 = 0$

$p(x) = (x-2)(x-3)(x-1)(x-2)(2x+5)$

$q(x) = 2x^5 - 13x^4 - 21x^3 + 127x^2 + 121x - 216$

Is $p(x) \equiv q(x)$?

Given $p(x), q(x)$ each of degree $\leq d$

are they the same?

Alg 1: Simplify each polynomial and compare coefficients... $O(d^2)$ (about d^2)

$r(x) = p(x) - q(x)$

Alg 2: Choose any set of $d+1$ numbers x_i, check if $r(x_i) = 0$ $\forall x_i$ $O(d^2)$

$1, \ldots, R$

Alg 3: Pick a random $x_0 \in [1, \ldots, R]$ Evaluate $R(x_0)$

IF $p(x) \neq q(x)$ Prob$(r(x_0) = 0) \leq \frac{d}{R}$

Repeat k times (successive independent random choices)

$\left(\frac{d}{R}\right)^k$ $O(kd)$

Quicksort

QUICKSORT(A):

- Choose pivot uniformly at random
- Partition $A \rightarrow A_1, p, A_2$

Return QUICKSORT(A_1), p, QUICKSORT(A_2)

$T(n)$: Expectation(# comparisons)

$T(1) = 0$

$T(2) = 1$

$\vdots$

$$T(n) = \frac{1}{n} \sum_{0}^{n-1} [T(k) + T(n-1-k)] + (n-1)$$

$$= (n-1) + \frac{2}{n} \sum_{0}^{n-1} T(k)$$

need to guess upper bound of $T(n)$

Guess: $T(n) \in \alpha n \log n$ for some constant α

Verify:

$T(n) = (n-1) + \frac{2}{n}\sum_{0}^{n-1} T(n)$

$\leq (n-1) + \frac{2}{n}\sum_{0}^{n-1} \alpha k \log k$

$\leq (n-1) + \frac{2}{n}\int_{0}^{n} \alpha k \log k$

$\leq (n-1) + \frac{2\alpha}{n}\left[\frac{n^2 \log n}{2} - \frac{n^2}{4} + \frac{1}{4}\right] = \alpha n \log n - \frac{\alpha n}{2} + \frac{\alpha}{2n}$

$= \alpha n \log n + \alpha n\left(\frac{1}{2} - \frac{1}{2n}\right) + \left(\frac{\alpha}{2n} - 1\right) < \alpha n \log n, \ \alpha > 2$

A: $a_1 < a_2 < a_3 \ldots a_j < a_n$

Input: Permutation of these numbers

$1 \leq i < j \leq n$ $X_{ij} = \begin{cases} 1, & \text{if } a_i \text{ and } a_j \text{ are compared w/ each other} \\ 0, & \text{o.w.} \end{cases}$

$\text{Prob}(X_{ij} = 1) = \frac{2}{j-i+1}$

$X = \sum_{i<j} X_{ij}$ X: r.v. # comparisons in running QS

What is $E(X)$?

Principle of linearity of expectation:

If x, y are r.v.'s
then $E(x+y) = E(x) + E(y)$

So...

$E(X) = \sum_{i<j} E(X_{ij}) = \sum_{i<j} \frac{2}{j-i+1}$

$= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{1}{k} \cdot \frac{2}{k} = \sum_{k=2}^{n} \frac{2}{k}(n-k+1)$

$= \sum_{2}^{n} \frac{2}{k}(n+1) - 2$

$= 2(n+1)\sum_{2}^{n} \frac{1}{k} - 2(n-1)$

$= 2nH_n - 2n - 2n + 2$

$= 2nH_n + 2 \simeq 1.44 n \log n - 4n + O(\log n)$

Lemma: suppose successive coin tosses are independent and that $p(\text{heads}) \geq p$ on each toss
let X = # tosses up to (including) the first head

$E(X) = \sum_{k=0}^{\infty} k \cdot \text{prob}(k-1 \text{ tosses are tails and } k^{th} \text{ is heads})$

$= \sum_{0}^{\infty} k(1-p)^{k-1} p$ *Let $q = 1-p$

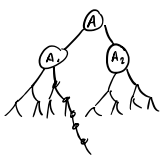
$= p\sum_{0}^{\infty} k q^{k-1}$

$= p\sum_{0}^{\infty} \frac{d}{dq} q^k$

$= p \frac{d}{dq}\left(\sum_{0}^{\infty} q^k\right)$

$= p \frac{d}{dq}\left(\frac{1}{1-q}\right) = p \cdot \frac{1}{(1-q)^2} = \frac{p}{p^2} = \frac{1}{p}$

$E(X) \leq \frac{1}{p}$



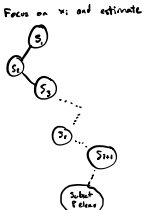
9/26

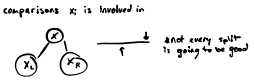
QUICKSORT:

To sort set S of N elems

1. If $|S| \leq 8$ do all $\binom{8}{2}$ comparisons
2. Pick random pivot $x \in S$
3. Partition $S \to S_L, x, S_R$
4. Recursively quicksort S_L, S_R

Focus on x_i and estimate # comparisons x_i is involved in





Split $S_j \to S_{j+1}$ is a "success" if
$|S_{j+1}| \leq \frac{7}{8}|S_j|$

Prob(success) $\geq 3/4$

successful splits $\leq \log_{8/7} n/8$

$E(\#\text{ splits between consec. successes}) \leq 4/3$

$E(\#\text{ splits to } x_i) \leq \frac{4}{3}\log_{8/7} n/8$

$E(\#\text{ comp involving } x_i) \leq \frac{4}{3}\log_{8/7} n/8 \approx 6 \log_2 n$
only with plenty of pivots when finally at 8 elem group

$E(\text{total \# comparisons}) \leq \frac{1}{2} n \cdot \frac{4}{3}\log_{8/7} n/8 + n/8\binom{8}{2}$
$\approx 3.5 n \log_2 n$

On any root-leaf E(#-splits) ≈ 6 log n

Prob(branch has fewer than 6 log n successes in 24 log n splits)

$$= \sum_{k=1+18\log n}^{24\log n} \binom{24\log n}{k} \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{24\log n - k} \qquad \left(\frac{m}{k}\right)^k \le \binom{m}{k} \le \left(\frac{me}{k}\right)^k$$

* Chernoff bound (Lecture 1)

$$= \sum_{k=1+18\log n}^{24\log n} \left(\frac{24e\log n}{4k}\right)^k < \sum_{k=1+18\log n}^{24\log n} \left(\frac{24e\log n}{4\cdot 18\log n}\right)^k$$

$$< \sum_{k=1+18\log n}^{24\log n} \left(\frac{e}{3}\right)^k = O\left(\left(\frac{e}{3}\right)^{18\log n}\right) = O\left(n^{-2.4}\right)$$

Pr ($\exists i$: v_i is involved in $\ge 24\log n$ splits)

$\le n \cdot$ Pr (x_i involved in $\ge 24\log n$ splits)

$\le n\, O(n^{-2.4}) = O(n^{-1.4})$

2D Game ?

"HIDDEN LINE ELIMINATION"

INPUT: Set of line segments:
$s_1, \ldots, s_N$ and viewpoint

OUTPUT: 1-D Rendering of visible portions of segments

* Painters Algorithm
- Build binary partition

Binary autopartition

9/28

Auto-Binary Partition

1. Randomly permute segments → $s_1, s_2, \ldots s_N$ (worst case is n^2)
2. For each region with ≥ 2 segments in it:
   cut it with s_i, i=min index segment that cuts the region

Thm: E(#regions) = O(N log N)

intersections = $\sum_u \sum_{v \ne u} C_{uv}$

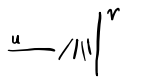
$E(\#\text{inter.}) = E\left(\sum_u \sum_{v \ne u} C_{uv}\right)$

$= \sum_u \sum_{v \ne u} E(C_{uv})$

Pr $(C_{uv} = 1) \le \frac{1}{\text{Index}(u,v)}$

$\le \sum_u \sum_{v \ne u} \frac{1}{\text{Index}(u,v)} = \sum_u \sum_{i=1}^{N-1} \frac{2}{i+1} \le 2N \sum_0^{N} \frac{1}{i+1}$

$= 2NH_N = O(N \log N)$

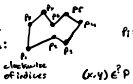


INDEX(u,v): # segments that $\ell(u)$ cuts before it reaches v (∞ if C_{uv} is 0)

Defn: $u \to v$: the $\ell(u)$ "cuts" segment v

$C_{uv} = \begin{cases} 1, & \ell(u) \text{ cuts } v \\ 0, & \end{cases}$

Simple Polygon:



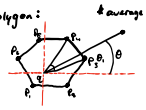
counter clockwise ordering of indices

$p_i = (x_i, y_i)$

$(x,y) \in P$

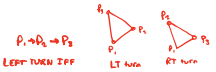
Convex Polygon:

* average 3 points to quickly get point inside polygon



$z \to q \to p_{i+1}$ RT turn
$z \to q \to p_i$ LT turn
z is outside

$p_i \to p_{i+1} \to z$ LT turn
z is inside



$p_1 \to p_2 \to p_3$
LEFT TURN IFF

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \text{ is}$$

LT turn

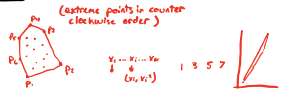
RT turn

Computing Convex Hull

INPUT: Set of points $S = \{p_1, p_2, \dots p_N\}$ in 2-dimensions

OUTPUT: Convex Hull of S.

(extreme points in counter clockwise order)

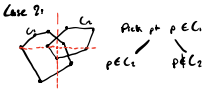


Divide and Conquer Idea:

1. Divide $S \to S_1, S_2$ each $N/2$ points
2. Rec. compute $CH(S_1)$, $CH(S_2)$
3. Merge $CH(S_1)$, $CH(S_2) \to CH(S)$

Case 1:
One CH completely in other

Case 2:



$C(N) \le 2C(\frac{N}{2}) + O(N)$

$O(N \log N)$

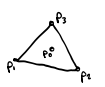
Simpler Alg:

10/3

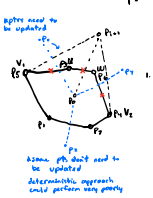
Grow the convex hull 1 pt at a time

INPUT: Set S of N pts in 2D. [No 3 pts are collinear]

1. Randomly permute S : $p_1, p_2, \dots p_N$
2. Let $S_3 = \{p_1, p_2, p_3\}$ and let $CH(S_3)$ be the C.H. of S_3 (in counter clockwise order)
3. Let p_0 be the centroid of S_3 (will always be in C.H.)

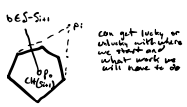


1. Maintain circular list of $CH(S_i)$
   $S_i = \{p_1, \dots, p_i\}$
2. For every pt in $S - S_i$, ptr from p to edge e of S_i that $\overline{pp}$ crosses
   (points not yet considered)



1. look at next pt, p_{i+1}
   a. If p_{i+1} lies inside CH_i, do nothing
   b. p_{i+1} lies outside $CH(S_i)$
   Let $v_1 v_2$ be edge of $CH(S_i)$ that $\overline{p_0 p_{i+1}}$ crosses

$CH(S_{i+1}) =$



can get lucky or unlucky with where we throw out what work we will have to do

Backwards run: deleting points

think backwards
remove $p \in S_i$
update edges $\to$
the point is that
# of updates forward
and backwards is the same

1. Pick random pt and delete it

$P($ptr v_1 needs updated$)$: 1 of the 2 pts running up being gets deleted

$P($ptr for $p \in S - S_i$ is updated$) \le \frac{2}{i-3} = O(\frac{1}{i})$

$E(\#\text{ ptr updates}) = O(\frac{n}{i})$

$E(\#\text{ ptr updates in Total}) = O(\sum_{i=1}^{n} (\frac{n}{i})) = O(n \sum \frac{1}{i}) = O(n \log n)$

PRIME TESTING

INPUT: A number $N \le 10^{100}$ (# bits $= \log N$)
if 100 digit number $N \le 10^{100}$

OUTPUT: { Yes: N is prime / No: N is not prime

Approach 1: Brute force

- Try dividing n by
  $2, 3, \dots \sqrt{n}$
  (all primes)

$\#$ divisions $\sim \frac{\sqrt{n}}{\log n} \sim 10^{50}$

$\#$ approx $\frac{n}{\log n}$ primes when $n \sim \dots$

probabilistic alg can say not prime or probably prime

Fermats Little Thm.

If p is prime then $\forall a < p$: $a^{p-1} \equiv 1 \pmod p$

$p = 7$
$a = 3$
$3^6 \equiv ? \pmod 7$
$3^2 \equiv 2$ "
$3^4 \equiv 4$ "
$3^6 \equiv 4 \cdot 2 \equiv 1 \pmod 7$

$p = 6$
$a = 2$
$2^{6-1} \equiv ? \pmod 6$ ⇒ witness that p is composite

for some p:
pick some a's and try and find witnesses
If not, p is probably prime

Carmichael Numbers:

⇒ composite #'s that $\forall a$ pass the F-test
$N = 3 \cdot 11 \cdot 17 = 561$
$\forall a < 561$ $a^{560} \equiv 1 \pmod{561}$

just b/c you can't find an "a" for a p that $a^{p-1} \not\equiv 1 \pmod p$, doesn't prove primality

Wilson's theorem:

p prime IFF $(p-1)! \equiv -1 \pmod p$

(very expensive)

Fact: $x^2 \equiv 1 \pmod p$ $\quad Z_p^* = \{1, \dots, p-1\}$

How many solutions in Z_p^* to this congruence when p is prime?

$x^2 - 1 \equiv 0 \pmod p$

$x^2 - 1 = ap$

$(x-1)(x+1) = ap \quad x = 1, -1$

*p cannot divide some x that is less than p, so $x = 1, -1$ are the only 2 sols

Exactly 2 solutions

CRT:

$x \equiv a_1 \pmod{p_1}$

$x \equiv a_2 \pmod{p_2}$

$\gcd(p_1, p_2) = 1$

Then there is a unique solution mod $p_1 p_2$ that satisfy these 2 congruences

what about when p is not prime?

p=15

satisfied by 1, -1, 4, 11

p = qr s.t. q, r are prime

$x^2 \equiv 1 \pmod q \pm 1$

$y^2 \equiv 1 \pmod r \pm 1$

(1,1) at least 4 square

(-1,-1) roots of unity

(1,-1) when p is not prime

(-1,1)

IDEA: try and find square roots of unity that aren't ± 1 quickly:

if we find one, number is composite, & this square root of unity is a witness

if not we might have been unlucky

$x^2 \equiv 1 \pmod{77}$

$(1,1) \xrightarrow{7^{11}} 1$

$(-1,-1) \to p-1 = 76$

$(1,-1) \to 43$

$(-1,1) \to 34$

Randomly trying: $4/77$

not efficient, need a more deliberate alg

Given p:

Check if even and 2:

IF odd:

pick random a

compute $a^{p-1} \equiv 1 \pmod p$

if not true, composite

else:

compute $a^{\frac{p-1}{2}} = 1, -1$

if not true, composite

else:

" $a^{\frac{p-1}{4}}$ "

recurse

*once -1 is hit, you stop

*if a number that isn't 1 or -1 is found, number is composite

10/5

FLT: p is prime $\Rightarrow \forall a \in \mathbb{Z}_p^* : a^{p-1} \equiv 1 \pmod p$

Fact: p is prime IFF $x^2 \equiv 1 \pmod p$ has exactly 2 solutions

INPUT: p.

1. If p is even and not 2 then REJECT (if p=2 then ACCEPT)
2. Select $a \in Z_p^*$ uniformly at random
3. Compute $a^{p-1} \bmod p$. If $\neq 1$, then REJECT
4. Let $p-1 = s \cdot 2^k$ where s is odd
5. Compute $a^{s \cdot 2^0}, a^{s \cdot 2^1}, \dots, a^{s \cdot 2^k}$ modulo p
6. If any element in sequence is $\neq 1$, then if the last elmt before 1 is NOT -1, REJECT
7. ACCEPT.

$Pr(\text{ACCEPT} \mid p \text{ is prime}) = 1$

$Pr(\text{ACCEPT} \mid p \text{ is composite}) \leq \frac{1}{2}$

Let d be a non-witness

Claim: $d \cdot t$ is a witness

$d^{s \cdot 2^j} \equiv \pm 1 \pmod p$

$d^{s \cdot 2^{j+1}} \equiv 1 \pmod p$

$(dt)^{s \cdot 2^j} = d^{s \cdot 2^j} \cdot t^{s \cdot 2^j}$

$\equiv \pm 1 \cdot ? \not\equiv \pm 1$

$(dt)^{s \cdot 2^{j+1}} \equiv 1 \pmod p$

$\to dt$ is a witness

non witness witnesses

Bfocd to show atleast injective

$d_1 \neq d_2 \Rightarrow d_1 t \neq d_2 t$

$d_1 t = d_2 t$

$\Rightarrow d_1 t \cdot t^{(k-2)(s-1)} = d_2 t \cdot t^{(k-2)(s-1)}$

$\Rightarrow d_1 t^{s \cdot 2^{k-1}} = d_2 t^{s \cdot 2^{k-1}}$

$\Rightarrow d_1 = d_2$

$t = 1 + q^{s-1}$

$t^2 = (1 + q^{s-1})^2 \equiv 1 + q q^{s-1} + (\dots)$ (all multiples of p)

$\equiv 1 \pmod p$

d: non-witness $\Rightarrow dt$ witness

$d^{p-1} \equiv 1 \pmod p$

$(dt)^{p-1} \equiv 1 \pmod p$