CS170 cribsheet midterm1

Order of Growth

Formal

UpperBound O : LowerBound Ω : Constant Θ
$\frac{a(n)}{b(n)} > 0, a(n) \in \Omega(b(n))$ $\frac{a(n)}{b(n)} < c, a(n) \in O(b(n))$
$\frac{a(n)}{b(n)} = c, a(n) \in \Theta(b(n))$

Tricks

$7^{\log(n)^2} = (2^{\log(7)})^{(\log(n))^2} = (2^{\log(n)})^{\log(7)\log(n)} \approx n^{\log(n)}$
$n! = 2^{n\log(n)}$ ::$36^5 = 6^{10}$
Solve the comparison by integration.
$(a+bi)*(c+di) \rightarrow r = ab, s = bd, t = (a+b)(c+d) = r - s + (t - r - s)i$

add/multiply

Karatsuba's $= \Theta(n^{\log_2 3})$

Prove

Geom sum series: $g(n) = \frac{1-c^{n+1}}{1-c} = \frac{c^{n+1}-1}{c-1}$
Induction: $\gcd(F_{k+1}, F_{k+1}) = \gcd(F_{k+1}, F_{k+2} - F_{K+1}) = \gcd(F_{k+1}, F_k) = 1$
Numbers before prime $1/n$: in $O(n)$ time. Geom dist.$E[X] = \sum_{i=1}^{\infty} i * P[X = i] = \sum_{i=1}^{\infty} i * (1-p)^{(i-1)}p$
$p = probheads, i - 1 = tailsthrows$
$= p * dp/dt(\sum_{i=1}^{\infty} -(1-p)^i) \rightarrow_{sums} = -1/p$ Integrate: $E[X] = p * (1/p^2) = 1/p$
Binary Search: if N is a square. Why only $\log n$ for power max? $N = q^k \rightarrow \log N = k\log \rightarrow k = \log N/\log q \leq \log N$
For any power: poweringoperation$\{\sum_{i=1}^{k} in * n = O(k^2 n^2)\}$
Repeat $\log n$ times to get $O(n^6)$

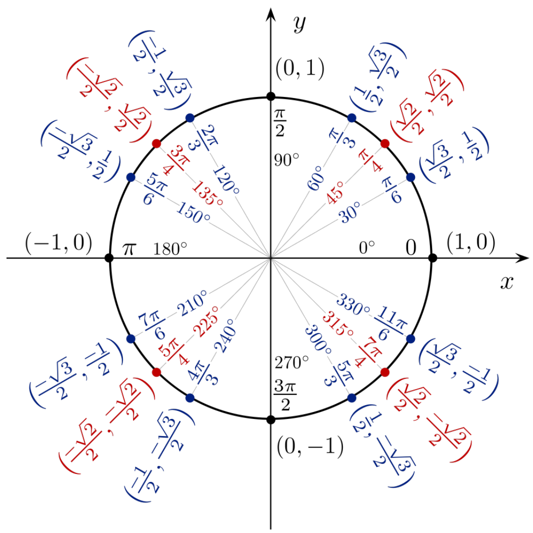
Modular Arithmetic

Quadratic residue busniess. Fermat's theorem:
$\forall 1 \leq a < p : a^{p-1} \equiv 1 mod p$ if p is prime.
Euler's Theorem: $m^{(p-1)(q-1)} \equiv 1(modpq)$ Multitudes:
$2013^{2014} = 3^{2012+2} = (3^{503})^4 * 3^2 = 1 * 3^2 = 4allmod5$
$2012^{2013} = 2^{2012+1} = (2^{503})^2 * 2^1 = 1 * 2 = 2allmod5$

$5^{170^{70}} mod 5$: take $170^{70} = 4s + t$ form
$170^{70} = (2*85)^{(2*35)} = (4*85^2)^{35} = 0mod4$
Worst RSA: We know N,e,d: $k = (ed-1)/(p-1)(q-1)$, limit k$\in 1, 2$ by $e = 3, d < (p-1)(q-1)$ Solve two eq system for p and q modulating k, use $N = pq$.
Randomize recoverable RSA w/ $(M^e * k^e)^d modN = MkmodN$ then multiply by k^{-1}
Primality testing: Doesn't catch Carmichaels. you did this for euler project already

Divide and Conquer

Master's Theorem:
$T(n) = aT(n/b) + O(n^d), a > 0, b > 1, d \geq 0$
$O(n^d) \rightarrow d > \log_b a$ :: $O(n^d \log n) \rightarrow d = \log_b a$ ::
$O(n^{\log_b a}) \rightarrow d < \log_b a$
Majority Element: If there is a majority element then it will be a majority element of A_1 or A_2, , $O(n\log n)$. Or you could use the pairing-discard approach $T(n) = T(n/2) + O(n) = O(n)$
Closest pair of points: ugh...



Complex number practice: $\omega = e^{2\pi i/8}, n = 8, = \sqrt{2}/2 + i\sqrt{2}/2$

$\omega^7 = e^{2\pi i(7/8)} = \sqrt{2}/2 - i\sqrt{2}/2 = \omega^-1, \omega^7 + \omega = \sqrt{2}$
$p(x) = x^2 + 1, p(\omega) = 1 + i, p(\omega^2) = 0, p(\omega^3) = 1 - i$

Missing integer: Array A of numbers [0,N]. Split into N/2 and count the bits in least significant position. You know how may 1-bits to expect. If that number is spot on, missing=0, otherwise missing=1. For each of these splits and counts we downsize by N/2 $\rightarrow T(n) = T(n/2) + O(n) = O(n)$, all without bit complexity

Pareto points: Sort $O(n\log n)$ and then do linear scan in reverse order $O(n)$

FFT: $A(x) = 1 + 2x - x^2 + 3x^3$
$(x_1, x_2, x_3, x_4) = (\omega^0, \omega^1, \omega^2, \omega^3) = (1, i, -1, -i)$ :: $\omega = e^{2\pi i/n}$
In general find the nearest power of two as n

Split into
$A(x) = A_e(x) + xA_o(x)$ :: $A_e(x) = 1 - x, A_o(x) = 2 + 3x$
$A_e(\omega^{2j}) + \omega^i A_o(\omega^{2j})$

DFT Matrix entry: $(m, n) = \omega^{m*n} = e^{(2\pi i/n)*mn}$ Inverse
DFT AMmtrix entry: $(m, n) = (1/n) * \omega^{-m*n} = e^{-(2\pi i/n)*mn}$

Graphs

Facts

Undirec graph w/ n verts and n edges has cycle by induction.

Stongly connected: path between any two points :: TREE EDGES <=> CROSS EDGES depending on DFS

2013 Zack Field