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.
$(a+bi)*(c+di) \to r = ab, s = bd, t = (a+b)(c+d) = r - s + (t - r - s)i$

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.$E[X] = \sum_{i=1}^{\infty} i * P[X = i] = \sum_{i=1}^{\infty} i*(1-p)^{(i-1)}p$
$p = probheads, i - 1 = tailsthrows$
$= p * dp/dt(\sum_{i=1}^{\infty} -(1-p)^i) \to_{sums} = -1/p$ Integrate: $E[X] = p*(1/p^2) = 1/p$
Binary Search: if N is a square. Why only $\log n$ for power max? $N = q^k \to \log N = k\log \to k = \log N/\log q \le \log N$
For any power: poweringoperation$\{\sum_{i=1}^{k} in * n = O(k^2n^2)\}$
Repeat $\log n$ times to get $O(n^6)$

Modular Arithmetic

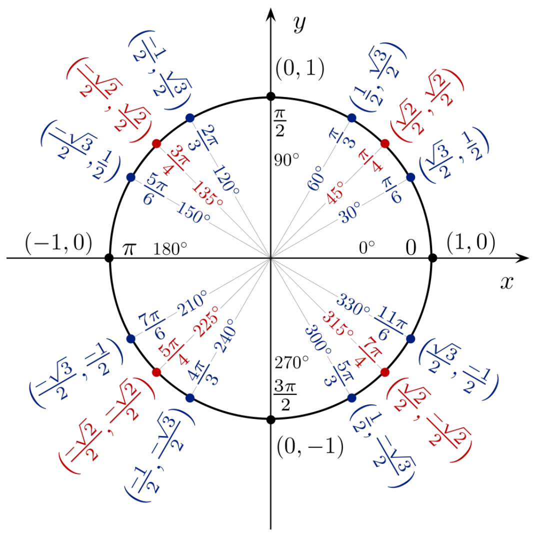
Quadratic residue busniess. Fermat's theorem:
$\forall 1 \le a < p : a^{p-1} \equiv 1 mod p$ if p is prime.
Euler's Theorem: $m^{(p-1)(q-1)} \equiv 1(modpq)$ Multitudes:
$2013^{2014} = 3^{2012+2} = (3^{503})^4 * 3^2 = 1 * 3^2 = 4 all mod 5$
$2012^{2013} = 2^{2012+1} = (2^{503})^2 * 2^1 = 1*2 = 2 all mod 5$
$5^{170^{70}} mod 5$: take $170^{70} = 4s + t$ form
$170^{70} = (2*85)^{(2*35)} = (4*85^2)^{35} = 0 mod 4$
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/ $(M^e * k^e)^d mod N = M k mod N$ then multiply by k^{-1}
Primality testing: Doesn't catch Carmichaels. you did this for euler project already

Divide and Conquer

Master's Theorem:
$T(n) = aT(n/b) + O(n^d), a > 0, b > 1, d \ge 0$
$O(n^d) \to d > \log_b a$ :: $O(n^d \log n) \to d = \log_b a$ ::
$O(n^{\log_b a}) \to d < \log_b a$
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 $T(n) = T(n/2) + O(n) = O(n)$

For finding kth smallest element in array, $O(n) average, O(n^2) worst$

$$\text{selection}(S,k) = \begin{cases} \text{selection}(S_L,k) & \text{if } k \le |S_L| \\ v & \text{if } |S_L| < k \le |S_L| + |S_v| \\ \text{selection}(S_R, k - |S_L| - |S_v|) & \text{if } k > |S_L| + |S_v|. \end{cases}$$



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 $\to T(n) = T(n/2) + O(n) = O(n)$, 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: $(m,n) = (1/n)*\omega^{-m*n} = e^{-(2\pi i/n)*mn}$

$A[1] = A(\omega^1) = A(\omega) = A_(\omega^2) + \omega A_o(\omega^2)$ Since
$FFT[a_0, a_2] = [A_e(\omega^0), A_e(\omega^2)]$and $FFT[a_1, a_3] = [A_o(\omega^0), A_o(\omega^2)]$
we read off the values, $A_e[\omega^2] = 2, A_o[\omega^2] = 1$from the given values.

Substituting, we get

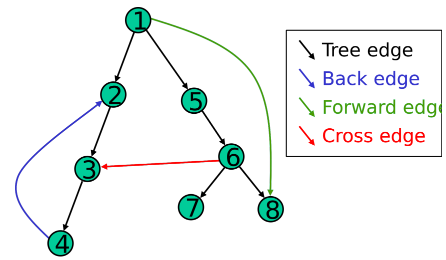
$A[1] = 2 + \omega = 2 + i$ (since $\omega = i$).

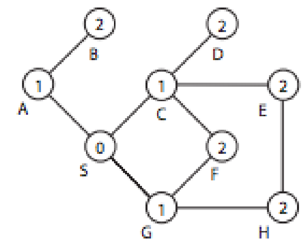
$$FFT([u,v,x,y])_k = FFT([u,x])_k + \omega^k FFT([v,y])_k$$
$$FFT([u,v,x,y])_{k+2} = FFT([u,x])_k - \omega^k FFT([v,y])_k,$$

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
Dijkstra's: Put all edges on a list and mark distance ∞, $O(|V|)$ time.





- Use convolution: This allows us to compute the score the pattern s_1 along each position s_2 in $O(n\log n)$ time.
- Change the alphabet from $\{0,1\}$ to $\{-1,1\}$. The reason for this is that matches will be scored as $(-1)^2 = 1^2 = 1$, but not equal to mismatches which are $(-1)1 = -1$.
- Notice that convlution does a flip and shift operation. However, we don't want the flip. We only want the shift. Thus, the solution is just to flip s_1 beforehand.

Using these three obervations, we now sketch out our solution. First, we define pattern s_1' of length n so that $s_1'(i) = -1$ if $s_1(m-i) = 0$, $s_1'(i) = 1$ if $s_1(m-i) = 1$ (the $m-i$ reverses the string for us) and $s_1'(i) = 0$ if $i > m$ (padding). Similarly, set s_2' to be s_2 , with all the zero bits replaced by -1. Now we seek to compute the convolution

$$c(i) = \sum_{k=1}^{i} s_1'(i) \cdot s_2'(i-k)$$

We can compute this convolution via an FFT, a point-wise product and an inverse FFT. Notice that the only interesting region of c is between $m-1 \le i \le n-1$ so we only need to check those indices (this is because any thing outside of this interval will try to match s_1 outside the boundary of s_2). Notice that if we have k matches, our sum at $c(i)$ will be $m-2k$, thus, if in the resulting solution we get any $c(i) \ge m-2k$ for $i \ge k$, then we have found a solution where the matched pattern is $s_2[i-m+1:i]$.

The running time of this algorithm, $O(n\log n)$, is dominated by the FFT and inverse FFT steps, which each take $O(n\log n)$ time. The point-wise product and search for $c(i) \ge m-2k$ each take $O(n)$ time.

2013 Zack Field