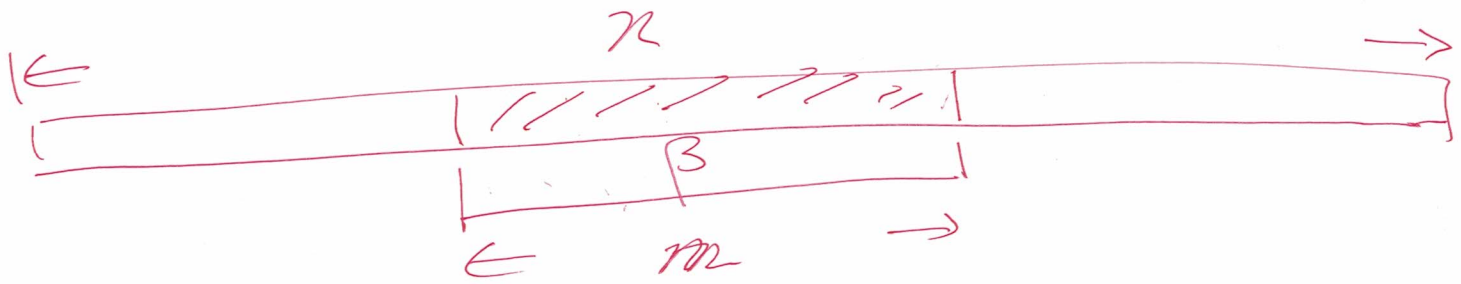
Cpts 515. 10/30/2020.

Last time: Karp-Rabin Alg.



$h(\underbrace{\alpha[i..m+i-1]}_{a}) \stackrel{?}{=} h(\underbrace{\beta}_{b})$.

ε: collision prbabty. We control the total error prbabty $n\varepsilon \leq 1/2$. .... (**).

In above, a and b are m-bit unsigned numbers.

$h(a) = h(b) \Rightarrow \underbrace{|a-b| \bmod q = 0}_{*}$.

all possible values of $|a-b|$.

↳ how many: 2^m values.

So, $|a-b|$ has at most m prime divisors.

how many g's to make (*) true? at most many.

We need find N s.t. $0..N$ contains $2nm$ primes. so that the error ε will be

$$\leq \frac{m}{2nm} = \frac{1}{2n}$$. to make (**) true.

How big is N?

Define $\pi(N) =$ "the # of primes $\leq N$".

From Euler, $\pi(N) \sim \frac{N}{\ln N}$.

Boter result (Rosser & Schoafield): when $N \geq 17$,

$$\frac{N}{\ln N} \leq \pi(N) \leq 1.25506 \frac{N}{\ln N}.$$

Now, $2nm \leq \frac{N}{\ln N}$.

$\Rightarrow N = (20nm)\ln(20nm)$.

↳ This is small!

Midterm

Linear algebra. ——— linear transform! but this is not the most useful topic for CS.

Matrix = Graph

Graph: ① path count (up to ∞ length)

② "density is a graph"

③ graph similarity / graph isomorphism.

⋮

Let G be a Strongly Connected Graph (assuming that G is one big SCC) with n nodes. M is G's adjacent matrix:

$M[i, j] = 1$ if node $i \rightarrow$ node j is an edge;

$M[i, j] = 0$ if " " " is not an edge.

---

(1) M has a unique and largest eigenvalue, called Perron number, denoted by λ. (a positive real #).

(2) All other eigenvalues of M are $< \lambda$;
(i.e., $|\lambda'| < \lambda$ for all other eigenvalues λ').
$\|\cdot\|$ norm of λ'.

A tool in your toolbox:

A is $m \times m$ matrix. A is positive. (≥ 0)

Eigenvalue λ: $A\Sigma = \lambda\Sigma$.

you have n eigenvalues.

left eigenvector $uA = \lambda u$.

right eigenvector $Av = \lambda v$.

MatLab can take care of these.

Next,

① $M^R =_{def} \underbrace{M \cdot M \cdot M \cdot \ldots \cdot M}_{n \text{ times}}$.

② $M^n[i,j] =$ the total # of walks from node i to node j in G;

③ M^n can be approx. by

$$\lambda^n \; V \; U^T$$

- λ: Perron number
- V: left eigenvector of λ
- U^T: Transpose of right eigenvector of λ

④ the total # of walks from node i to node j with length n, $M^n[i,j]$, can be approx. by

$$\frac{v_i u_j}{\|u\|} \cdot \lambda^n \cdot v u^T$$

where $\|u\| = \sum_k u_k$, and v_i, u_j are the components in vectors v, u.

⑤. the total # of walks from node j with length n, can be approx. by

$$\frac{v_j}{\|u\|} \cdot \lambda^n \cdot v u^T \quad \text{.....} (*)$$

Now

$$\limsup_{n\to\infty} \frac{\log |S_n|}{n} = \log \lambda \quad \text{using } (*)$$

λ: Perron number

where S_n : the total # of walks from v_0.

What if G has multiple SCC?

each SCC has a λ.

take max of all the λ's.

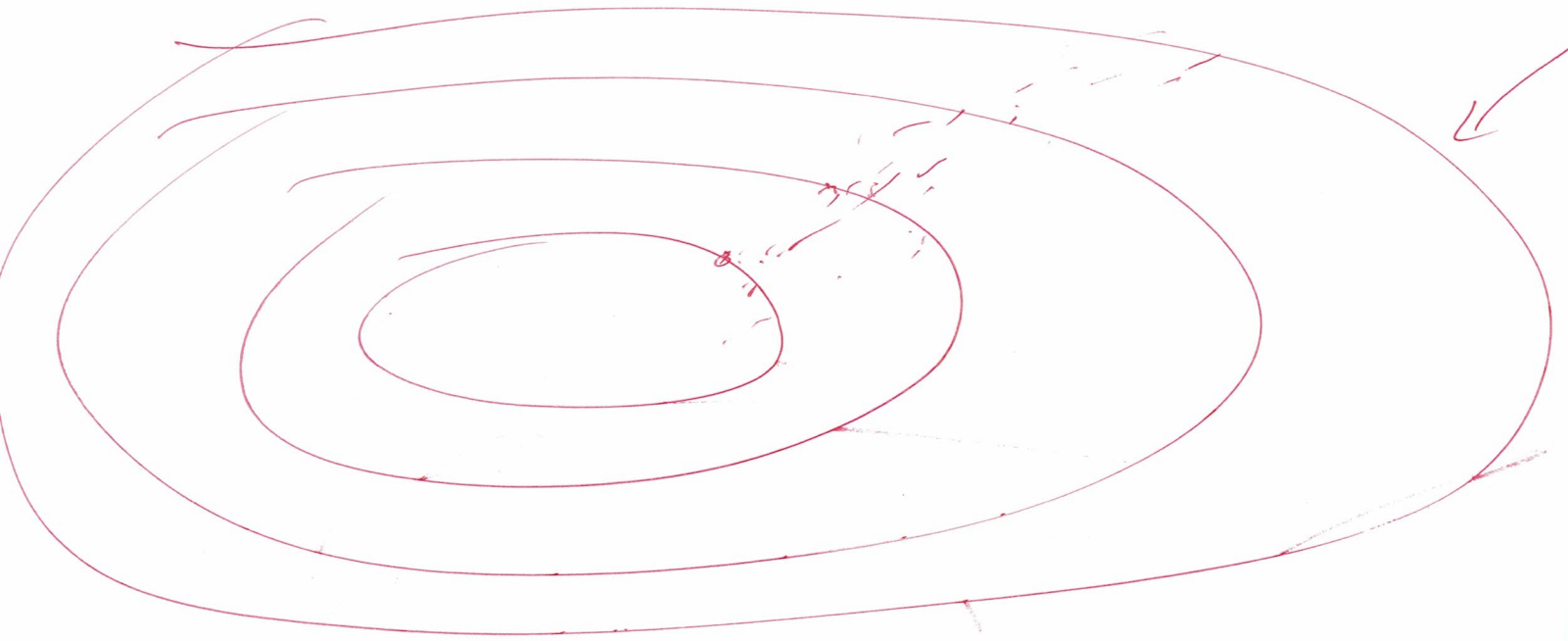
$$\limsup_{n\to\infty} \frac{\log |S_n|}{n} = \log \lambda$$

↑ We know how to get this.

For different applications, you need interpret the S_n differently:

① S_n is the # of walks with length $= n$.

② "walk" ——— exple.



energy: high $\Rightarrow$ more ways to walk
(# of orbits is high).