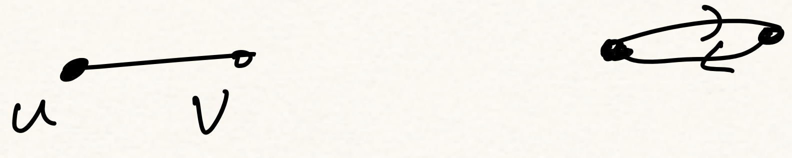
# Lecture 14 10/10/2025

## Random walks in undirected graphs

A finite state Markov chain corresponds to a random walk on a weighted directed graph. Random walks in undirected graphs have many stronger and nicer properties and a number of applications. They also are closely related reversible Markov chains.

Suppose $G=(V,E)$ is an undirected graph. We let $\vec{G}=(V,\vec{E})$ be the corresponding bidirected graph

$G$  $\vec{G}$

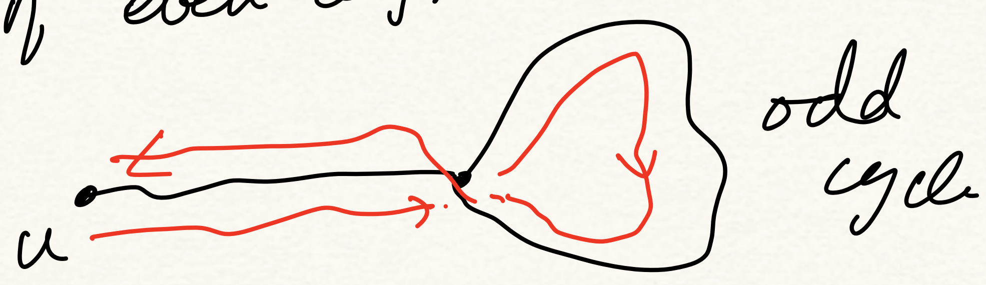
We can consider weights on the edges but for simplicity we assume all are 1 (we allow multi-graphs).

A random walk on $G$ is the following stochastic process. Start at some random vertex given by a probability distribution $\pi(0)$ on $V$. In each step, if we are at vertex $v$, pick a uniform random edge in $\delta(v)$ and and go to the end point of $v$.
Note that if the edge is a self-loop we stay at $v$.

We can think of this random walk as a Markov chain on $\vec{G}$ where each directed edge $(v,u)$ is given probability $\frac{1}{\deg(v)}$.

Lemma: Suppose $G$ is a loop-less connected graph. Then $G$ is aperiodic iff $G$ is not bipartite.

Proof: $G$ is bipartite $\Rightarrow$ underlying chain has period 2 since all cycles and closed walks have even length.
If $G$ is not bipartite $\Rightarrow$ $G$ has an odd length cycle. In $\bar{G}$ we have that each vertex is in a closed walk of even length and one with odd length.




$\Rightarrow$ period is 1.

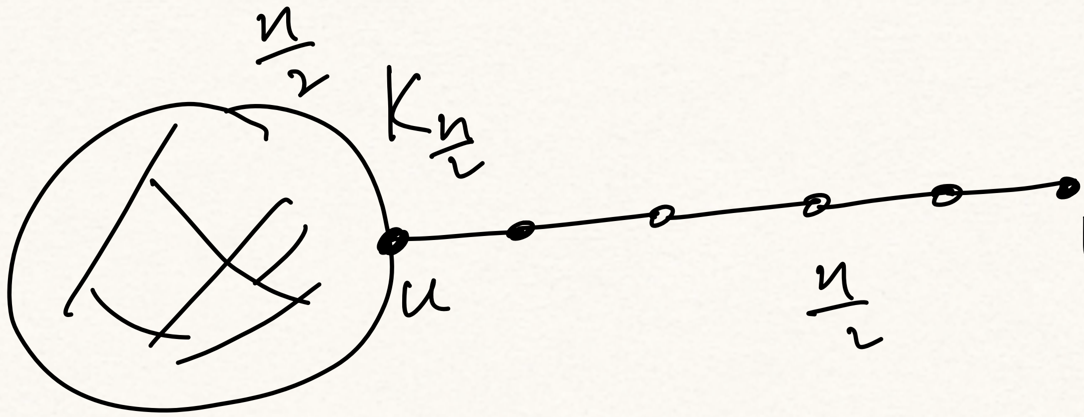
We can either assume $G$ is not bipartite or add self loops on each vertex and make the walk lazy. This will ensure walk is ergodic.

<u>Lemma</u>: A random walk on $G$ converges to a stationary distribution $\pi$ where $\pi_v = \frac{deg(v)}{2m}$.

Proof: Exercise. Verify that this satisfies $\pi P = \pi$ for the underlying Markov Chain.

Let $h_{u,v}$ be the expected time to reach state $v$ when starting at $u$.

Hitting time is not necessarily symmetric




Lollipop graph $L_n$

$h_{u,v} = \Theta(n^3)$ and $h_{v,u} = \Theta(n^2)$

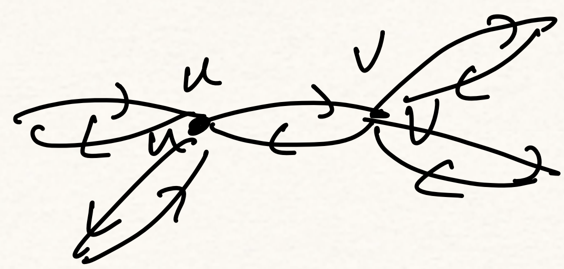
Also $L_n$ shows that adding edges can increase $h_{u,v}$ and $C_{u,v}$.

Commute time is $h_{u,v} + h_{v,u}$ which is symmetric.

We will prove two basic results using elementary methods.

Lemma: For any edge $uv \in E$
$h_{u,v} + h_{v,u} \leq 2m$.

Proof: Consider $\vec{G}$. We can view the random walk on $G$ as a random walk on $\vec{E}$. That is the state space is $\vec{E}$. Consider this chain.



Consider the transition matrix $Q$ for this chain. It turns out to be doubly stochastic:
$$Q_{(u,v),(v,w)} = \frac{1}{\deg(v)}.$$

For a normal transition matrix row sum is 1 but here column sum is also 1. Easy to verify.

$\Rightarrow$ $(1, 1, 1, , 1)$ is a left eigen vector of $Q$ $\Rightarrow$ by normalizing $\frac{1}{|\vec{E}|}$, the stationary distribution for $Q$ is $\left(\frac{1}{2m}, \ldots, \frac{1}{2m}\right)$ the uniform distribution.

$\Rightarrow$ $h_{(u,v),(u,v)} = 2m$ where $h_{(u,v),(u,v)}$ is the expected time in the edge-walk chain to start on ~~edge~~ arc $(u,v)$ and revisit $(u,v)$. We can interpret such a walk as giving an upper bound on $h_{u,v} + h_{v,u}$.

$h_{(u,v),(u,v)} \leq 2m \Rightarrow$ if the original random walk traversed the ~~ed~~ arc $(u,v)$ then the expected time to traverse $(u,v)$ again is $2m$.

But since original walk on $G$ is memoryless, once it reaches $v$ it shows that the expected time to visit $u$ and take edge $u \to v$ is at most $2m$.

But this walk is only one way to start at $v$ and reach $u$ and back to $v$ $\Rightarrow$ $h_{v,u} + h_{u,v} \leq 2m$.

$\square$

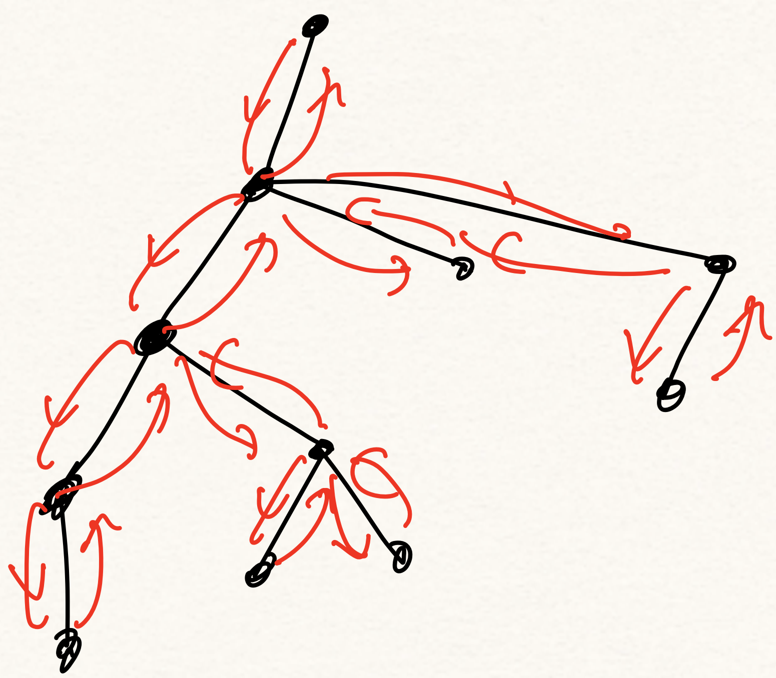
<u>Caveat:</u> Note above holds only for $uv \in E$. We will later see a more refined version when $u,v$ is not necessarily an edge.

Defn: The cover time of a graph $G = (V, E)$ is the max over all $v \in V$ of the expected time to visit all the vertices. $C_v$ is cover time starting at $v$.

$$C(G) = \max_{v \in V} C_v.$$

Theorem: $C(G) \leq 2m(n-1)$.

Proof: Consider a spanning tree $T$.



We can consider an Eulerian walk of $T$.

Say it is $v_1 \to v_2 \to v_3 \to \cdots \to v_{2n} = v_1$

We can upper bound $C(G)$ by

$$h_{v_1,v_2} + h_{v_2,v_3} + \cdots + h_{v_{2n-2},v_{2n-1}}$$

$$= \sum_{uv \in E(T)} (h_{u,v} + h_{v,u})$$

$$\leq 2m(n-1). \qquad \square$$

One can prove another interesting upper bound on cover time.

<u>Theorem</u>: $C(G) \leq H(n-1) \max_{\substack{u,v \in V \\ u \neq v}} h_{u,v}$.

## Applications

① s-t connectivity in $O(\log n)$ space.

Suppose we are given an undirected graph written on read-only memory in adjacency list / matrix format. We want to use very little extra memory to see if some given s can reach t. We can easily do this using $O(n)$ space by using graph search (BFS / DFS).

Can we do this with $O(\log n)$ space? Note that writing s, t takes $O(\log n)$ bits.

Yes if we allow randomization.

How? Start a random walk at $s$. Because $C(G) = O(n^3)$ if we don't see $t$ after $O(n^3 \log n)$ steps we know whp that $s$ is not connected to $t$!

Can implement random walk in $O(\log n)$ space.

$G$ can be bipartite so need to use lazy random walk. Doesn't change details too much.

# 2-SAT

Boolean formula $\phi$ on $x_1, x_2, \ldots, x_n$ where each clause has exactly 2 variables. Can check if

$$\phi = (x_1 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_3) \vee (x_2 \vee x_1).$$

2-SAT is solvable in P. How?

One nice way to see it is via random walks.

## Schonig's Alg:

1. Let $\bar{a} = a_1 a_2 \ldots a_n \in \{0,1\}^n$ be an arbitrary assignment to $x_1, \ldots, x_n$
2. While $\bar{a}$ does not satisfy $\phi$ do
   - Let $C_i$ be an arbitrary clause

that is not satisfied by $\bar{a}$.

- Pick a literal of $C_i$ uniformly at random
- Flip the assignment for the chosen literal ~~to~~ and update $\bar{a}$.

<u>Lemma</u>: If $\phi$ is satisfiable algorithm terminates in $O(n^2)$ steps.

<u>Proof</u>: Suppose $\bar{b}$ is a fixed satisfying assignment. Let $\bar{a}_t$ be the assignment ~~at~~ after $t$ steps. Let $x_t = dist(\bar{b}, \bar{a}_t)$ be the Hamming distance between $\bar{b}$ and $\bar{a}_t$. That is, the number of variables in which $\bar{a}_t$ differs from $\bar{b}$.

If $\alpha_t = 0$ then alg terminates.
$\alpha_t \leq n$. The algorithm can be viewed as doing a random walk on state space $(0, 1, 2, \ldots, n)$. and starting at position $dist(\bar{b}, \bar{a}_0)$

Since only one variable is changed distance changes - by $+1$ or $-1$.

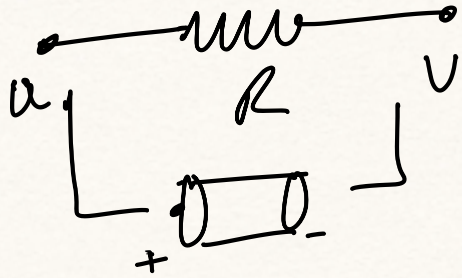
Since $C_i$ is picked as an unsatisfied clause, at least one literal is incorrect and hence with prob at least $\frac{1}{2}$ we will reduce distance.

Thus we can view walk on

$\{0,1,2,\dots,n\}$. Worst case is $\frac{1}{2}$ on each side. Can view it as random walk on a finite line. Cover time is $O(n^2)$. $\Rightarrow$ will visit 0 in $O(n^2)$ in expectation.

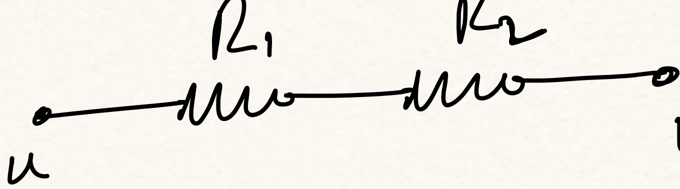
D.

# Electrical Networks and Random Walks

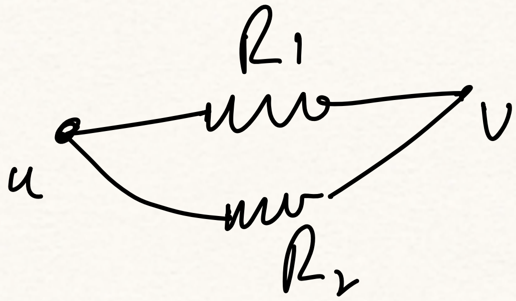



Ohm's law $V = IR$

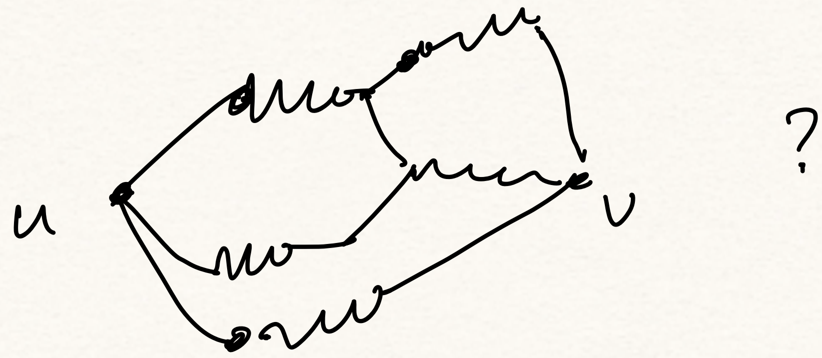
voltage = current times resistance.




effective resistance is $R_1 + R_2$




effective resistance is $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$.




Let $R_{u,v}$ be effective resistance.

<u>Theorem:</u> $\forall u,v \quad C_{uv} = h_{u,v} + h_{v,u}$
$$= 2m R_{u,v}.$$

<u>Corollary:</u> If $uv \in E$ then
$$C_{uv} \leq 2m.$$