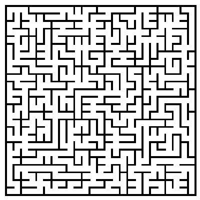
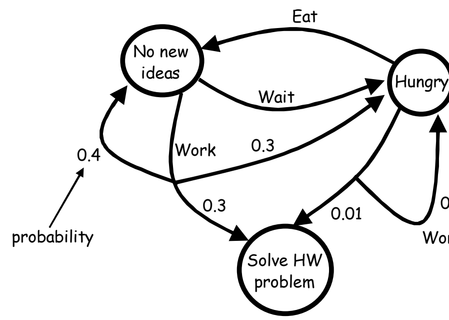
Great Theoretical Ideas In Computer Science
Anupam Gupta CS 15-251 Fall 2005
Lecture 23 Nov 14, 2005 Carnegie Mellon University

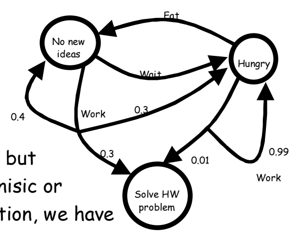
Random Walks



An abstraction of student life



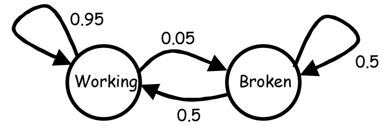
Markov Decision Processes



Like finite automata, but instead of a determinisic or non-deterministic action, we have a probabilistic action.

Example questions: "What is the probability of reaching goal on string Work,Eat,Work,Wait,Work?"

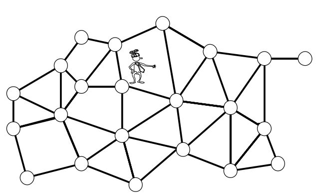
Even simpler models: Markov Chains



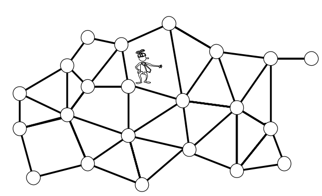
e.g. modeling faulty machines here.
No inputs, just transitions.

Example questions: "What fraction of time does the machine spend in repair?"

And even simpler: Random Walks on Graphs

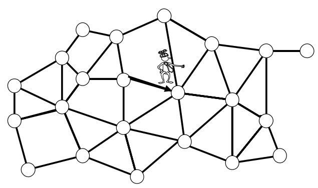


Random Walks on Graphs



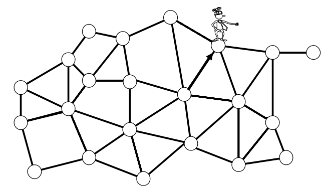
At any node, go to one of the neighbors of the node with equal probability.

Random Walks on Graphs



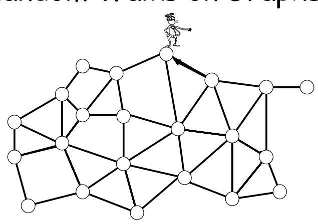
At any node, go to one of the neighbors of the node with equal probability.

Random Walks on Graphs



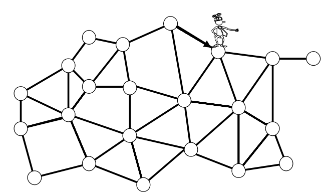
At any node, go to one of the neighbors of the node with equal probability.

Random Walks on Graphs

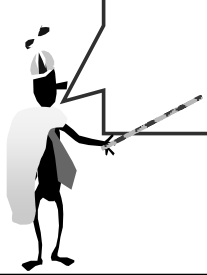


At any node, go to one of the neighbors of the node with equal probability.

Random Walks on Graphs



At any node, go to one of the neighbors of the node with equal probability.

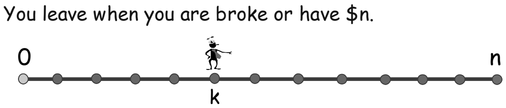


Let's start simple…

We'll just walk in a straight line.

Random walk on a line

You go into a casino with $k, and at each time step, you bet $1 on a fair game.
You leave when you are broke or have $n.

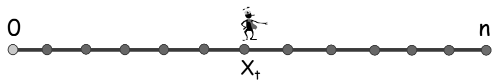


Question 1:
what is your expected amount of money at time t?

Let X_t be a R.V. for the amount of money at time t.

Random walk on a line

You go into a casino with \$k, and at each time step, you bet \$1 on a fair game.
You leave when you are broke or have \$n.

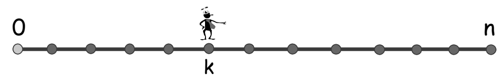


$X_t = k + \delta_1 + \delta_2 + ... + \delta_t$,
(δ_i is a RV for the change in your money at time i.)

$E[\delta_i] = 0$, since $E[\delta_i|A] = 0$ for all situations A at time i.
So, $E[X_t] = k$.

Random walk on a line

You go into a casino with \$k, and at each time step, you bet \$1 on a fair game.
You leave when you are broke or have \$n.



Question 2:
what is the probability that you leave with \$n ?

Random walk on a line

Question 2:
what is the probability that you leave with \$n ?

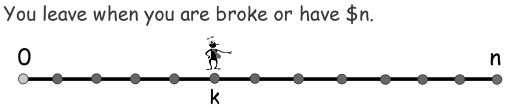
$E[X_t] = k$.

$E[X_t] = E[X_t \mid X_t = 0] \times \Pr(X_t = 0)$	0
$+ E[X_t \mid X_t = n] \times \Pr(X_t = n)$	$+ n \times \Pr(X_t = n)$
$+ E[X_t \mid \text{neither}] \times \Pr(\text{neither})$	$+ (\text{something}_t \times \Pr(\text{neither}))$

As $t \to \infty$, $\Pr(\text{neither}) \to 0$, also $\text{something}_t < n$
Hence $\Pr(X_t = n) \to k/n$.

Another way of looking at it

You go into a casino with \$k, and at each time step, you bet \$1 on a fair game.
You leave when you are broke or have \$n.

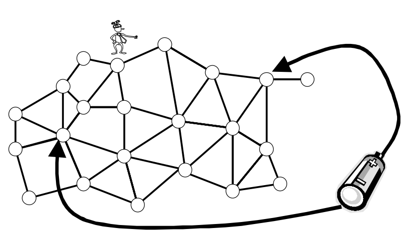


Question 2:
what is the probability that you leave with \$n ?

= the probability that I hit green before I hit red.

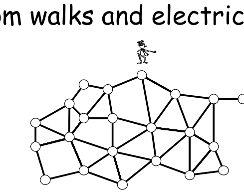
Random walks and electrical networks

What is chance I reach green before red?



Same as voltage if edges are resistors and we put 1-volt battery between green and red.

Random walks and electrical networks

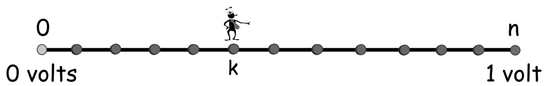


- p_x = Pr(reach green first starting from x)
- $p_{green} = 1$, $p_{red} = 0$
- and for the rest $p_x = \text{Average}_{y \in Nbr(x)}(p_y)$

Same as equations for voltage if edges all have same resistance!

Electrical networks save the day…

You go into a casino with $k, and at each time step, you bet $1 on a fair game.
You leave when you are broke or have $n.

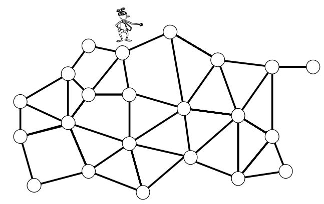


Question 2:
what is the probability that you leave with $n ?

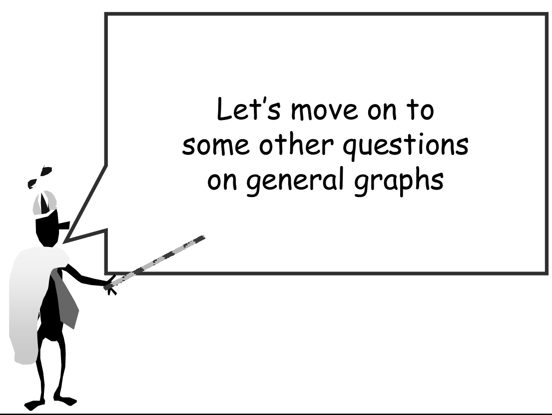
voltage(k) = k/n
= Pr[hitting n before 0 starting at k] !!!

Random walks and electrical networks

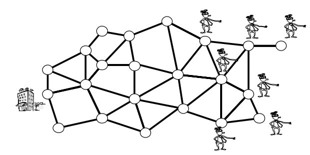
What is chance I reach green before red?



Of course, it holds for general graphs as well…



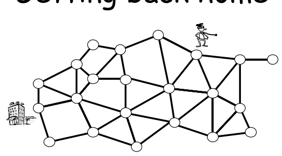
Getting back home



Lost in a city, you want to get back to your hotel.
How should you do this?

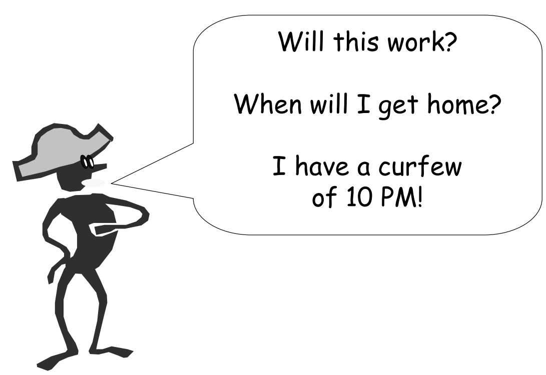
Depth First Search:
requires a good memory and a piece of chalk

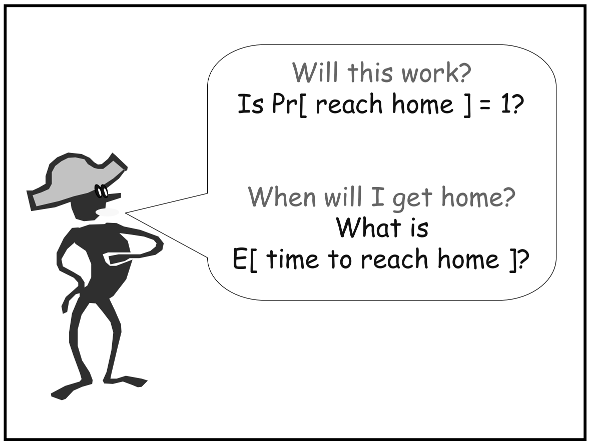
Getting back home

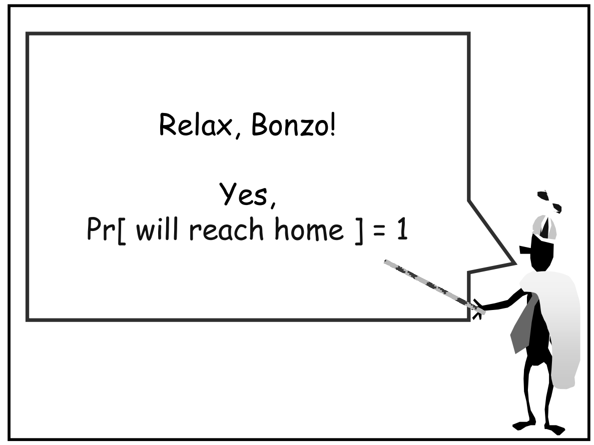


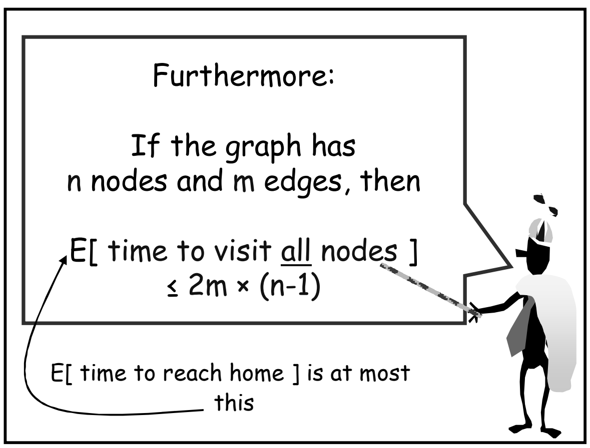
Lost in a city, you want to get back to your hotel.
How should you do this?

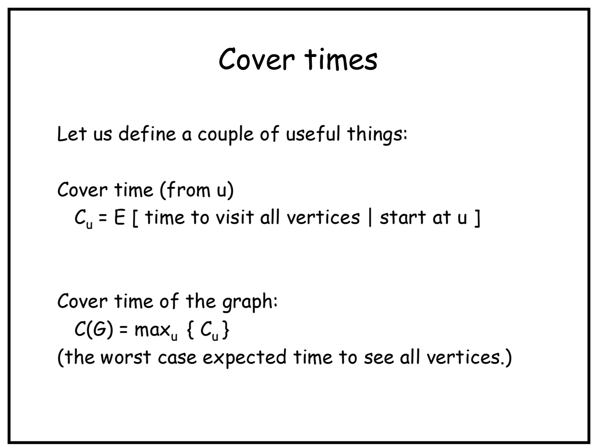
How about walking randomly?
no memory, no chalk, just coins…

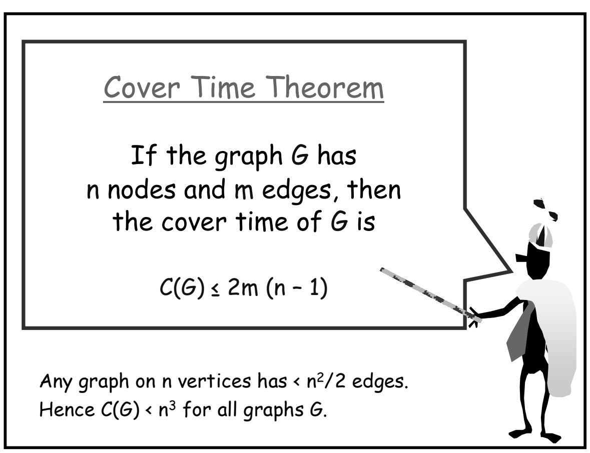


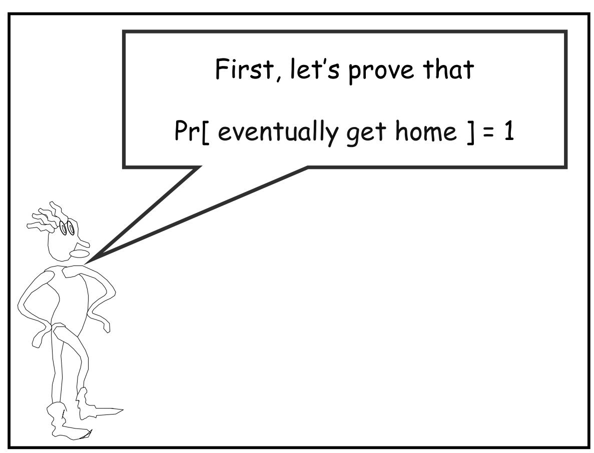

Will this work?
Is Pr[reach home] = 1?
When will I get home?
What is
E[time to reach home]?


Relax, Bonzo!
Yes,
Pr[will reach home] = 1


Furthermore:
If the graph has
n nodes and m edges, then
E[time to visit all nodes]
≤ 2m × (n-1)
E[time to reach home] is at most this


Cover times
Let us define a couple of useful things:
Cover time (from u)
C_u = E [time to visit all vertices | start at u]
Cover time of the graph:
$C(G) = \max_u \{ C_u \}$
(the worst case expected time to see all vertices.)


Cover Time Theorem
If the graph G has
n nodes and m edges, then
the cover time of G is
$C(G) \leq 2m (n - 1)$
Any graph on n vertices has < $n^2/2$ edges.
Hence $C(G) < n^3$ for all graphs G.

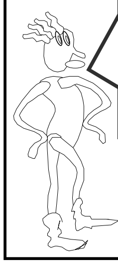

First, let's prove that
Pr[eventually get home] = 1

We will eventually get home

Look at the first n steps.
There is a non-zero chance p_1 that we get home.
Also, $p_1 \geq (1/n)^n$

Suppose we fail.
Then, wherever we are, there a chance $p_2 \geq (1/n)^n$ that we hit home in the next n steps from there.

Probability of failing to reach home by time kn
$= (1 - p_1)(1 - p_2) \ldots (1 - p_k) \to 0$ as $k \to \infty$



Actually, we get home pretty fast…

Chance that we don't hit home by $2k \times 2m(n-1)$ steps is $(\frac{1}{2})^k$

But first, a simple calculation

If the average income of people is $100 then more than 50% of the people can be earning more than $200 each

True or False?

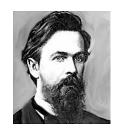
False! else the average would be higher!!!

Markov's Inequality

If X is a non-negative r.v. with mean E(X), then

$\Pr[\ X > 2\ E(X)\ ] \leq \frac{1}{2}$

$\Pr[\ X > k\ E(X)\ ] \leq 1/k$



Andrei A. Markov

Markov's Inequality

Non-neg random variable X has expectation A = E[X].

$A = E[X] = E[X \mid X > 2A]\ \Pr[X > 2A] + E[X \mid X \leq 2A]\ \Pr[X \leq 2A]$

$\geq E[X \mid X > 2A]\ \Pr[X > 2A]$ ← since X is non-neg

Also, $E[X \mid X > 2A] \quad > 2A$

$\Rightarrow A \geq 2A \times \Pr[X > 2A] \qquad \Rightarrow \frac{1}{2} \geq \Pr[X > 2A]$

Pr[X exceeds k × expectation] ≤ 1/k.

An averaging argument

Suppose I start at u.
E[time to hit all vertices | start at u] ≤ C(G)

Hence, by Markov's Ineq.
Pr[time to hit all vertices > 2C(G) | start at u] ≤ ½.

Why?
Else this average would be higher.

so let's walk some more!

Pr [time to hit all vertices > 2C(G) | start at u] ≤ ½.

Suppose at time 2C(G), am at some node v, with more nodes still to visit.

Pr [haven't hit all vertices in 2C(G) more time | start at v] ≤ ½.

Chance that you failed **both** times ≤ ¼ = $(\frac{1}{2})^2$!

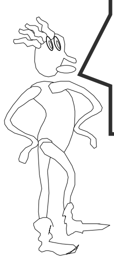
The power of independence

It is like flipping a coin with tails probability q ≤ ½.

The probability that you get k tails is $q^k \le (\frac{1}{2})^k$.
(because the trials are independent!)

Hence,
Pr[havent hit everyone in time k × 2C(G)] ≤ $(\frac{1}{2})^k$

Exponential in k!

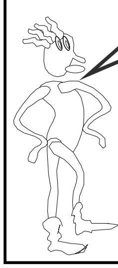


Hence, **if** we know that

Expected Cover Time
C(G) < 2m(n-1)

then

Pr[home by time 4k m(n-1)] ≥ 1 – $(\frac{1}{2})^k$



Now for a bound on the cover time of any graph….

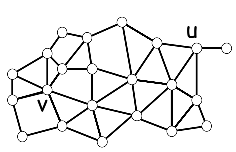
Cover Time Theorem

If the graph G has n nodes and m edges, then the cover time of G is

C(G) ≤ 2m (n – 1)

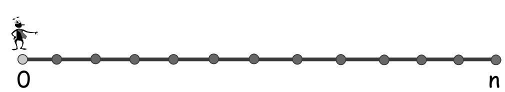
Electrical Networks again

"hitting time" H_{uv} = E[time to reach v | start at u]
Theorem: If each edge is a unit resistor
$H_{uv} + H_{vu} = 2m \times \text{Resistance}_{uv}$



Electrical Networks again

"hitting time" H_{uv} = E[time to reach v | start at u]
Theorem: If each edge is a unit resistor
$H_{uv} + H_{vu} = 2m \times \text{Resistance}_{uv}$

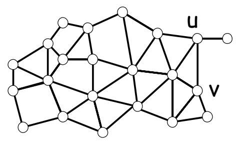


$H_{0,n} + H_{n,0} = 2n \times n$
But $H_{0,n} = H_{n,0} \Rightarrow H_{0,n} = n^2$

Electrical Networks again

"hitting time" H_{uv} = E[time to reach v | start at u]
Theorem: If each edge is a unit resistor
$H_{uv} + H_{vu} = 2m \times \text{Resistance}_{uv}$

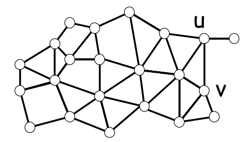
If u and v are neighbors $\Rightarrow$ $\text{Resistance}_{uv} \leq 1$
Then $H_{uv} + H_{vu} \leq 2m$



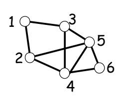
Electrical Networks again

If u and v are neighbors $\Rightarrow$ $\text{Resistance}_{uv} \leq 1$
Then $H_{uv} + H_{vu} \leq 2m$

We will use this to prove the Cover Time theorem
$C_u \leq 2m(n-1)$ for all u



Suppose G is this graph

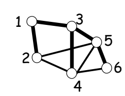


Pick a spanning tree of G

Say 1 was the start vertex,
$C_1 \leq H_{12}+H_{21}+H_{13}+H_{35}+H_{56}+H_{65}+H_{53}+H_{34}$
$\leq (H_{12}+H_{21}) + H_{13}+ (H_{35}+H_{53}) + (H_{56}+H_{65}) + H_{34}$

Each $H_{uv} + H_{vu} \leq 2m$, and we have (n-1) edges in a tree

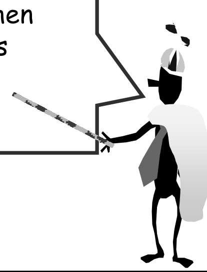
$C_u \leq (n-1) \times 2m$



Cover Time Theorem

If the graph G has
n nodes and m edges, then
the cover time of G is

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



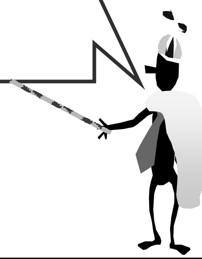
Hence, we have seen

The probability we start at x and hit Green before Red is
Voltage of x
if Voltage(Green) = 1, Voltage(Red) = 0.

The cover time of any graph is at most 2m(n-1).

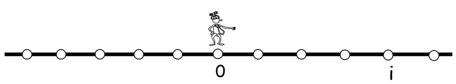
Given two nodes x and y, then
"average commute time" $H_{xy} + H_{yx} = 2m \times \text{resistance}_{xy}$

Random walks on infinite graphs



> A drunk man will find his way home, but a drunk bird may get lost forever
>
> *- Shizuo Kakutani*

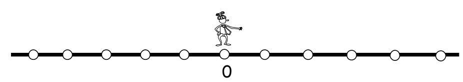
Random Walk on a line



Flip an unbiased coin and go left/right.
Let X_t be the position at time t

$\Pr[X_t = i]$
$= \Pr[\#heads - \#tails = i]$
$= \Pr[\#heads - (t - \#heads) = i] = \binom{t}{(t-i)/2} / 2^t$

Unbiased Random Walk



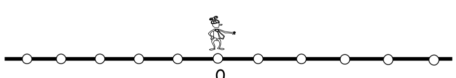
$\Pr[X_{2t} = 0] = \binom{2t}{t} / 2^{2t}$

Stirling's approximation: $n! = \Theta((n/e)^n \times \sqrt{n})$

Hence: $(2n)!/(n!)^2 = \dfrac{\Theta((\frac{2n}{e})^{2n}\sqrt{2n})}{\Theta((\frac{n}{e})^{n}\sqrt{n})}$

$= \Theta(2^{2n}/n^{\frac{1}{2}})$

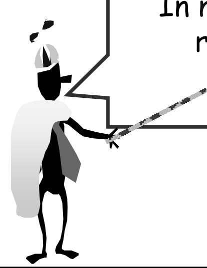
Unbiased Random Walk



$\Pr[X_{2t} = 0] = \binom{2t}{t} / 2^{2t} \le \Theta(1/\sqrt{t})$ ← Sterling's approx.

Y_{2t} = indicator for $(X_{2t} = 0)$ $\Rightarrow$ $E[Y_{2t}] = \Theta(1/\sqrt{t})$

Z_{2n} = number of visits to origin in 2n steps.
$\Rightarrow E[Z_{2n}] = E[\sum_{t=1\ldots n} Y_{2t}]$
$= \Theta(1/\sqrt{1} + 1/\sqrt{2} + \ldots + 1/\sqrt{n}) = \Theta(\sqrt{n})$



In n steps, you expect to return to the origin $\Theta(\sqrt{n})$ times!

Simple Claim

Recall: if we repeatedly flip coin with bias p
E[# of flips till heads] = 1/p.

Claim: If Pr[not return to origin] = p, then
E[number of times at origin] = 1/p.

Proof: H = never return to origin. T = we do.
Hence returning to origin is like getting a tails.
E[# of returns] =
E[# tails before a head] = 1/p – 1.
(But we started at the origin too!)

We will return…

Claim: If Pr[not return to origin] = p, then
E[number of times at origin] = 1/p.

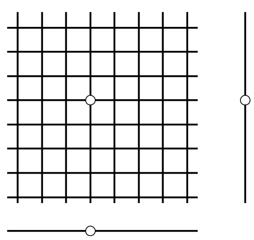
Theorem: Pr[we return to origin] = 1.

Proof: Suppose not.
Hence p = Pr[never return] > 0.
$\Rightarrow$ E [#times at origin] = 1/p = constant.

But we showed that E[Z_n] = $\Theta(\sqrt{n}) \rightarrow \infty$

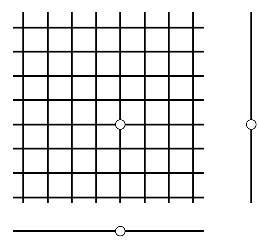
How about a 2-d grid?

Let us simplify our 2-d random walk:
move in both the x-direction and y-direction…



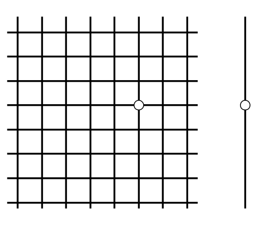
How about a 2-d grid?

Let us simplify our 2-d random walk:
move in both the x-direction and y-direction…



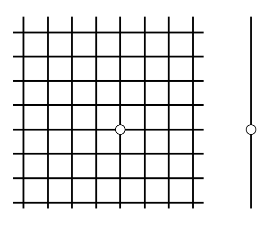
How about a 2-d grid?

Let us simplify our 2-d random walk:
move in both the x-direction and y-direction…



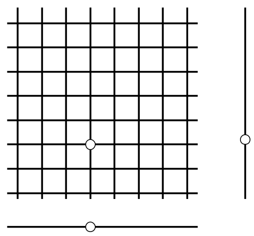
How about a 2-d grid?

Let us simplify our 2-d random walk:
move in both the x-direction and y-direction…



How about a 2-d grid?

Let us simplify our 2-d random walk:
move in both the x-direction and y-direction…



in the 2-d walk

Returning to the origin in the grid
⇔ both “line” random walks return to their origins

Pr[visit origin at time t] = $\Theta(1/\sqrt{t}) \times \Theta(1/\sqrt{t})$
= $\Theta(1/t)$

E[# of visits to origin by time n]
= $\Theta(1/1 + 1/2 + 1/3 + \ldots + 1/n) = \Theta(\log n)$

We will return (again!)…

Claim: If Pr[not return to origin] = p, then
E[number of times at origin] = 1/p.

Theorem: Pr[we return to origin] = 1.

Proof: Suppose not.
Hence p = Pr[never return] > 0.
⇒ E [#times at origin] = 1/p = constant.

But we showed that E[Z_n] = $\Theta(\log n) \to \infty$

But in 3-d

Pr[visit origin at time t] = $\Theta(1/\sqrt{t})^3 = \Theta(1/t^{3/2})$

$\lim_{n\to\infty}$ E[# of visits by time n] < K (constant)

Hence
Pr[never return to origin] > 1/K.

Much more fun stuff

Connections to electrical networks, and with eigenstuff

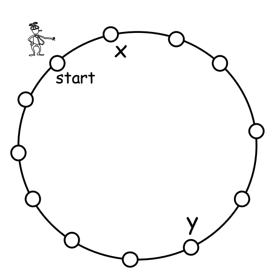
Applications to graph partitioning, random sampling, queueing theory, machine learning

Also, fun probabilistic facts.

A cycle game

Suppose we walk on the cycle till we see all the nodes.

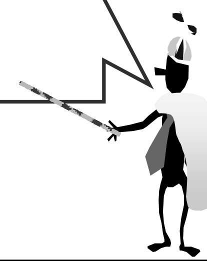
Is x more likely than y to be the <u>last</u> node we see?



But wait, there more…

The remaining stuff is optional.
If you want, please read on…

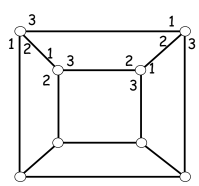
Let us see a cute implication of the fact that we see all the vertices quickly!



“3-regular” cities

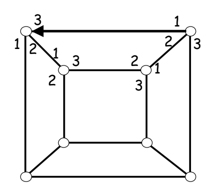
Think of graphs where every node has degree 3.
(i.e., our cities only have 3-way crossings)

And edges at any node are numbered with 1,2,3.



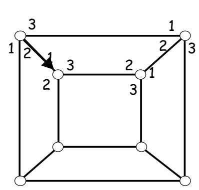
Guidebook

Imagine a sequence of 1’s, 2’s and 3’s
12323113212131…
Use this to tell you which edge to take out of a vertex.



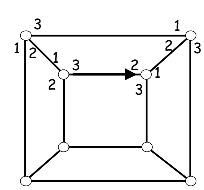
Guidebook

Imagine a sequence of 1’s, 2’s and 3’s
12323113212131…
Use this to tell you which edge to take out of a vertex.



Guidebook

Imagine a sequence of 1’s, 2’s and 3’s
12323113212131…
Use this to tell you which edge to take out of a vertex.

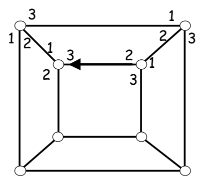


Guidebook

Imagine a sequence of 1's, 2's and 3's

12323113212131…

Use this to tell you which edge to take out of a vertex.



Universal Guidebooks

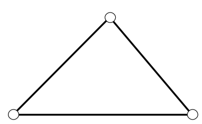
Theorem:

There exists a sequence S such that,
for all degree-3 graphs G (with n vertices),
and all start vertices,
following this sequence will visit all nodes.

The length of this sequence S is $O(n^3 \log n)$.

This is called a "universal traversal sequence".

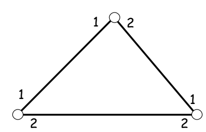
degree=2 n=3 graphs



Want a sequence such that
- for all degree-2 graphs G with 3 nodes
- for all edge labelings
- for all start nodes

traverses graph G

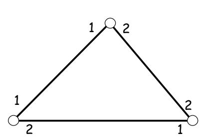
degree=2 n=3 graphs



Want a sequence such that
- for all degree-2 graphs G with 3 nodes
- for all edge labelings
- for all start nodes

traverses graph G

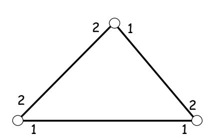
degree=2 n=3 graphs



Want a sequence such that
- for all degree-2 graphs G with 3 nodes
- for all edge labelings
- for all start nodes

traverses graph G

degree=2 n=3 graphs



Want a sequence such that
- for all degree-2 graphs G with 3 nodes
- for all edge labelings
- for all start nodes

traverses graph G

122

Universal Traversal sequences

Theorem:

There exists a sequence S such that for
- all degree-3 graphs G (with n vertices)
- all labelings of the edges
- all start vertices

following this sequence S will visit all nodes in G.

The length of this sequence S is $O(n^3 \log n)$.

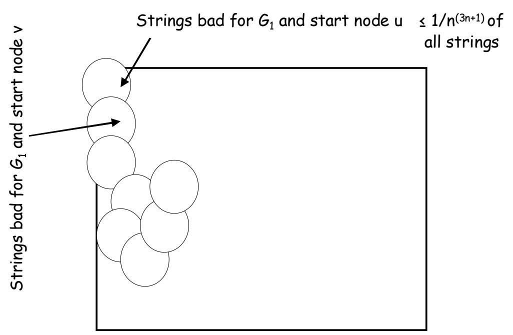
Proof

At most $(n-1)^{3n}$ degree-3 n-node graphs.
Pick one such graph G and start node u.

Random string of length 4km(n-1) fails to cover it with probability $\frac{1}{2}^k$.

If $k = (3n+1) \log n$, probability of failure $< n^{-(3n+1)}$

I.e., less than $n^{-(3n+1)}$ fraction of random strings of length 4km(n-1) fail to cover G when starting from u.



All length 4km(n-1) length random strings

Proof

How many degree-3 n-node graph are there?

For each vertex, specifying neighbor 1, 2, 3 fixes the graph (and the labeling).

This is a 1-1 map from
$\{\text{deg-3 n-node graphs}\} \rightarrow \{1\ldots(n-1)\}^{3n}$

Hence, at most $(n-1)^{3n}$ such graphs.

Proof (continued)

Each bite takes out at most $1/n^{(3n+1)}$ of the strings.

But we do this only $n(n-1)^{3n} < n^{(3n+1)}$ times.
(Once for each graph and each start node)

$\Rightarrow$ Must still have strings left over!
(since fraction eaten away $= n(n-1)^{3n} \times n^{-(3n+1)} < 1$)

These are good for every graph and every start node.

Univeral Traversal Sequences

Final Calculation:
This good string has length
4km(n-1)
$= 4 \times (3n+1) \log n \times 3n/2 \times (n-1)$.
$= O(n^3 \log n)$

Given n, don't know efficient algorithms to find a UTS of length n^{10} for n-node degree-3 graphs.

But here's a randomized procedure

Fraction of strings thrown away

$= n(n-1)^{3n} / n^{3n+1}$

$= (1 - 1/n)^n \rightarrow 1/e = .3678$

Hence, if we pick a string at random,
Pr[it is a UTS] > ½

But we can't quickly check that it is…

Aside

Did not really need all nodes to have same degree.
(just to keep matters simple)

Else we need to specify what to do, e.g.,
if the node has degree 5 and we see a 7.

References and Further Reading

Doyle and Snell, Random Walks and Electrical Networks
http://front.math.ucdavis.edu/math.PR/0001057

Motwani and Raghavan
Randomized Algorithms, Cambridge Univ Press.

Alon and Spencer
The Probabilistic Method, John Wiley & Sons.