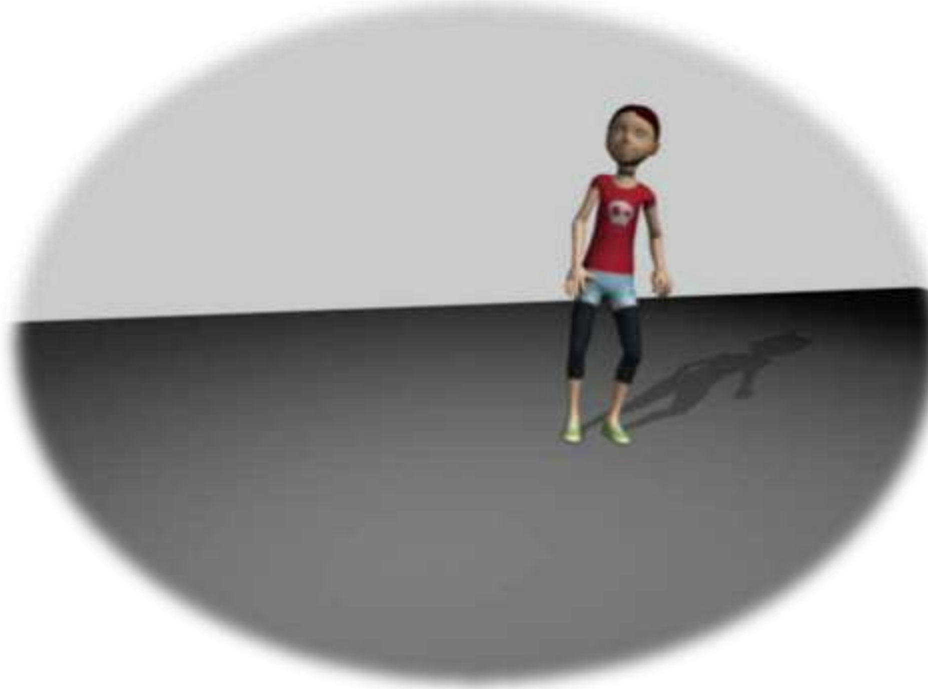
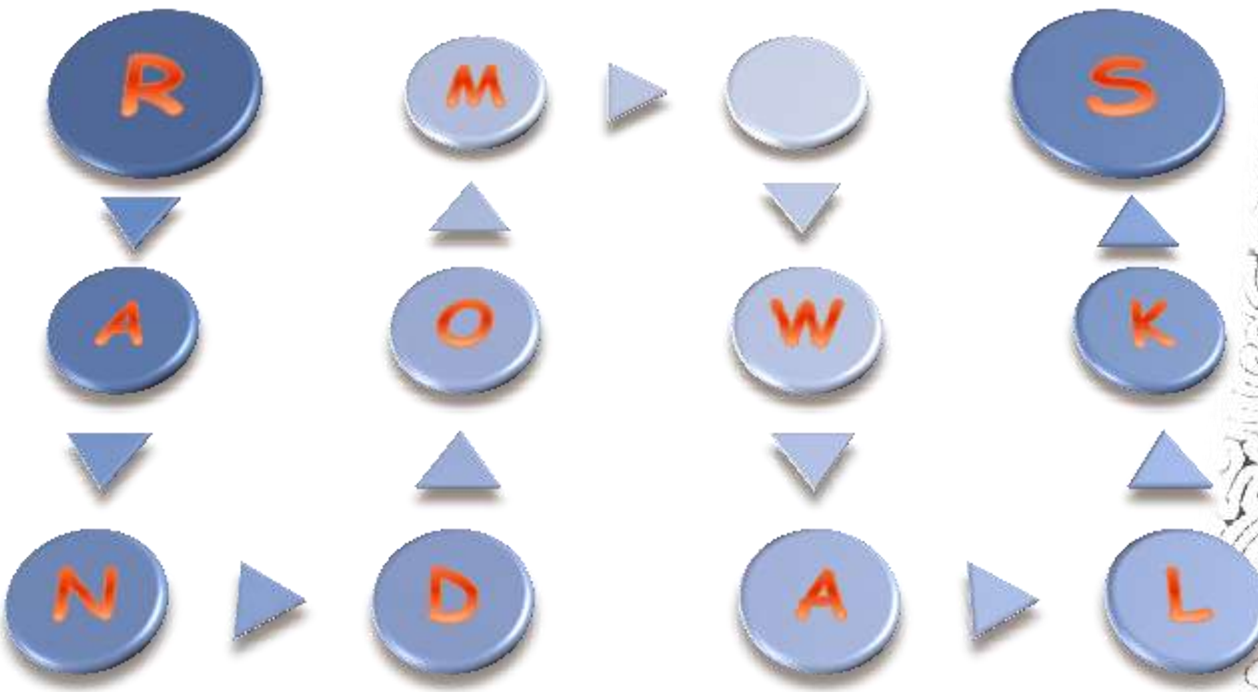
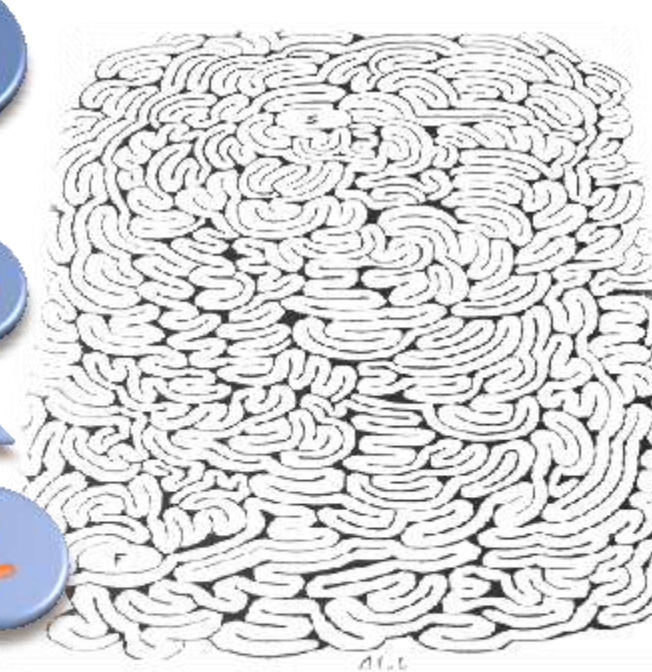
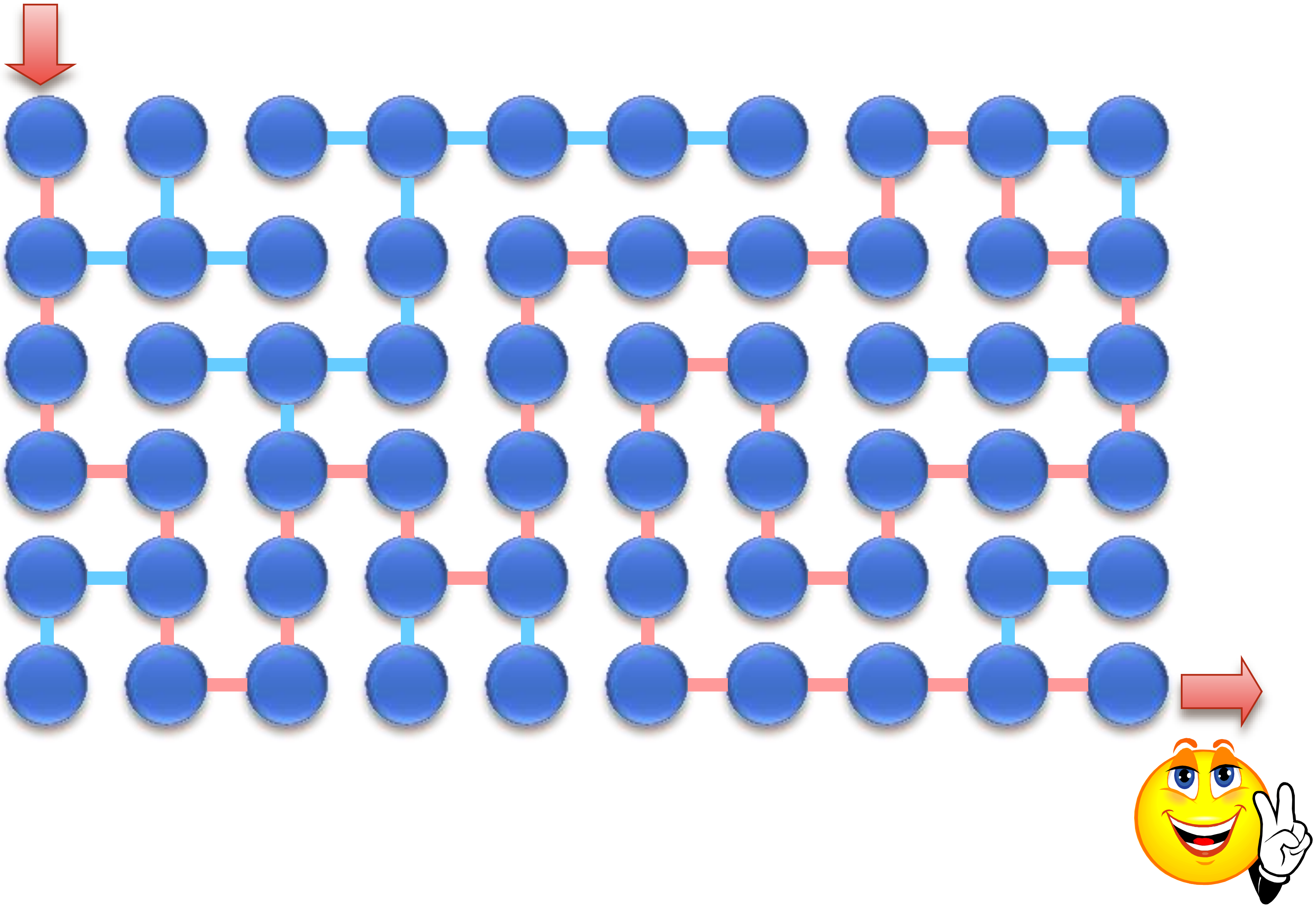



R
A
N
D
O
M
W
A
L
K
S



Is there a Solution?



Goal

- Undirected Connectivity
- In Random LOGSPACE

Plan

- Introduce Random Walks

Undirected Connectivity

Instance:

- An undirected graph $G=(V,E)$ and two vertices $s,t \in V$

Decision Problem:

- Is there a path in G from s to t

Theorem:

- CONN $\in$NL

Proof:

- Nondeterm. walk

What shall we do for an undirected graph?

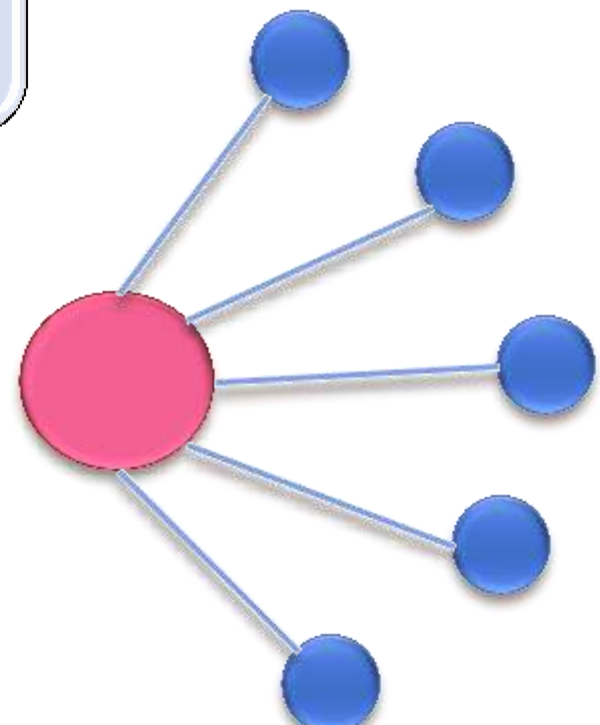
Nondeterministic vs. Random

All-powerful **guesses**

- which **neighbor** to go to **next**

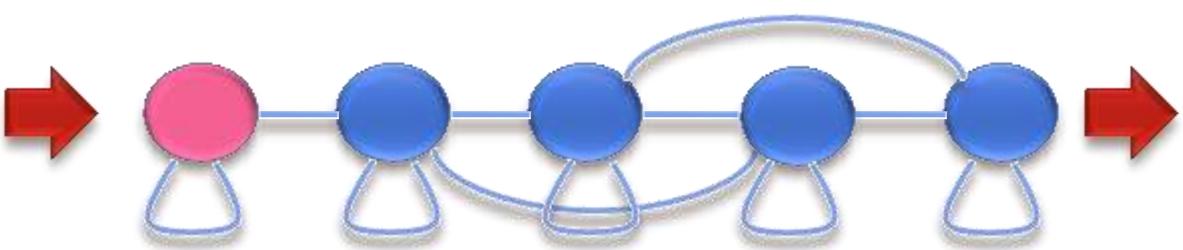
Randomly **guess**

- which **neighbor** to go to **next**



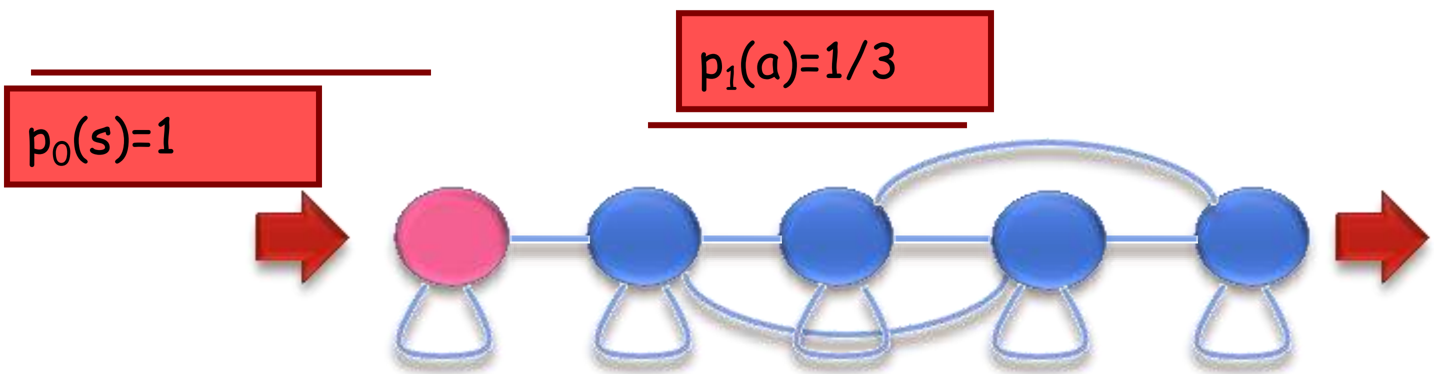
Random Walks

0	• Add a self loop to each vertex
Start	• at s
Let	• d_i be the degree of the current node.
Jump	• to each neighbor with probability $1/d_i$
Stop	• if reach t



Notation:

- Let v_n denote the vertex visited at time n (v_0=s)
- Let $p_n(i) = Pr[v_n=i]$



Stationary Distribution

Lemma:

- Let $G=(V,E)$ be a connected graph, then for any $i \in V$

$$\lim_{n\to\infty}[p_n(i)] = \frac{d_i}{2|E|}$$

Weaker Claim

Lemma:

- If for some **n**, for all **i∈V**

$$p_n(i) = \frac{d_i}{2|E|}$$

then for all **i∈V**

$$p_{n+1}(i) = \frac{d_i}{2|E|}$$

Proof:

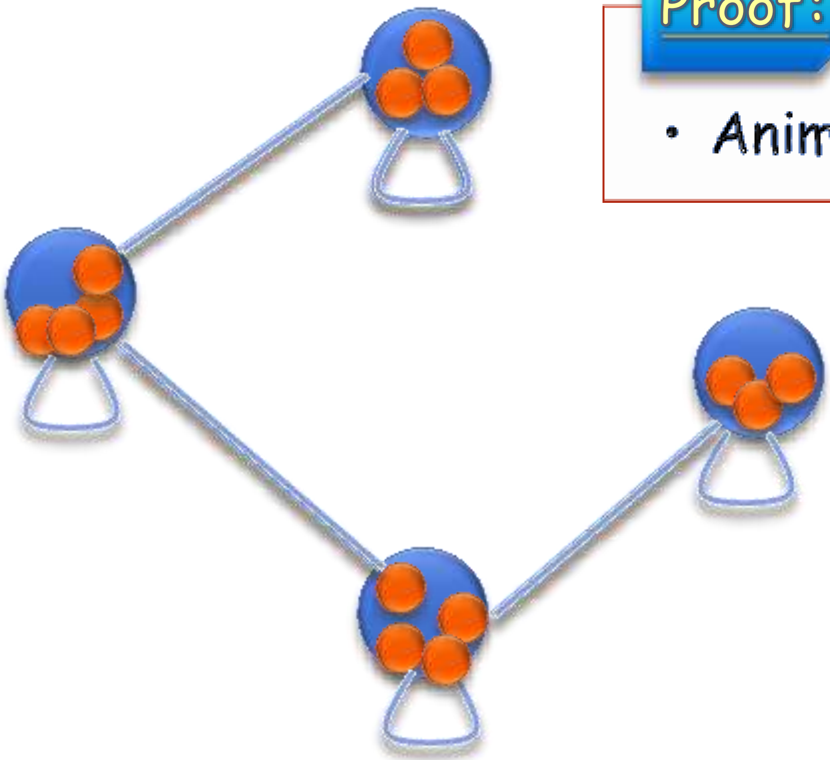
FixPoint

- $\Sigma d_i = 2|E|$

Lemma: Vertex i has probability d_i at time n ⇒ Vertex i has the same probability at time $n+1$ ■

Proof:

- Animated



Proof:

FixPoint

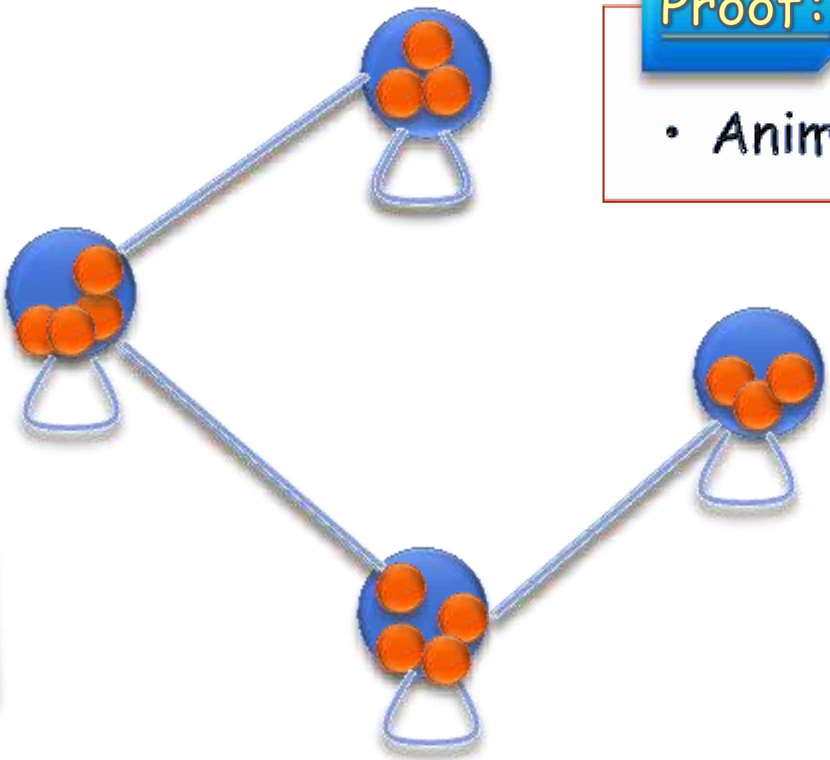
- $\Sigma d_i = 2|E|$

  Lemma: Vertex i has probability d_i at time n ⇒ Vertex i has the same probability at time $n+1$ ■

Proof:

- Animated

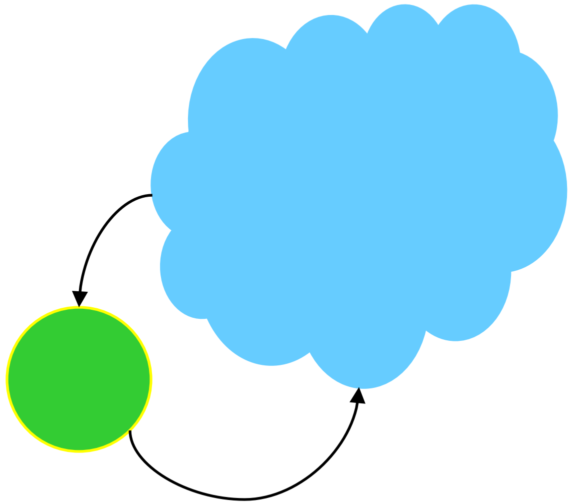
Note: every edge has same going both directions



Using the Asymptotic Estimate

Corollary:

- Starting from any vertex i, the *expected* time till revisit i is $2|E|/d_i$



Proof:

- If expected time were longer, vertex i would have lower probability in the stationary distribution ■

Otherwise: at limit walk were to spend <1 step out of any $2|E|/d_i$ at i

One-Sided Error

Note

- Always returns 'NO' if right answer is 'NO'

Hence

- the algorithm has one-sided error

RL

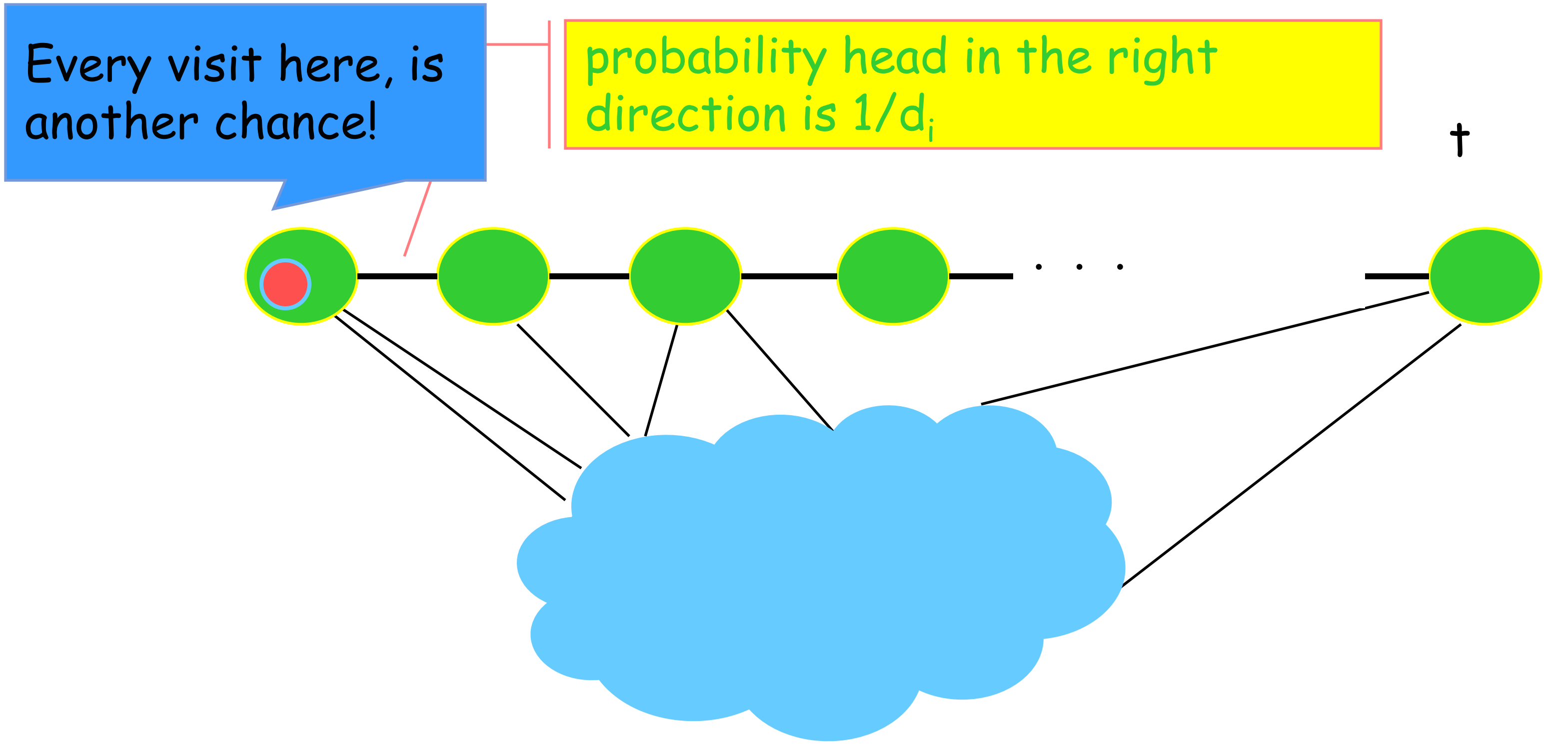
- is the class when limited to log space

Such algorithms are sometimes called "Monte-Carlo" algorithms

Number of Steps Required?

Note:

- If there is a path, in how many steps do we expect to arrive at t?



How Many Steps Expectedly Needed?

Since

- *expectedly* the walk returns to vertex i within $2|E|/d_i$ steps

It *expectedly*

- heads along the path within $2|E|$ steps

By linearity of expectation

- it is *expected* to reach t within $d(s,t) \cdot 2|E| \leq 2|V| \cdot |E|$ steps.

RL Algorithm for Undirected CONN

Run

- random walk from s

halt and *accept*

- if vertex t is visited

halt and *reject*

- after $2c \cdot |V| \cdot |E|$ steps

∃a path from s to t

Probably no path from s to t

true -by Markov- w/probability 1/c

Sum Up

Theorem:

- The above algorithm uses logarithmic space

Log space to maintain a pointer

On 'NO' instances

- always right

Markov: $Pr(X > c \cdot E[X]) < 1/c$

for 'YES' instances

- errs with probability at most $1/c$

Summary

explored
- undirected connectivity problem.

showed
- log-space randomized algorithm for this problem.

introduced
- an important technique: random walks.