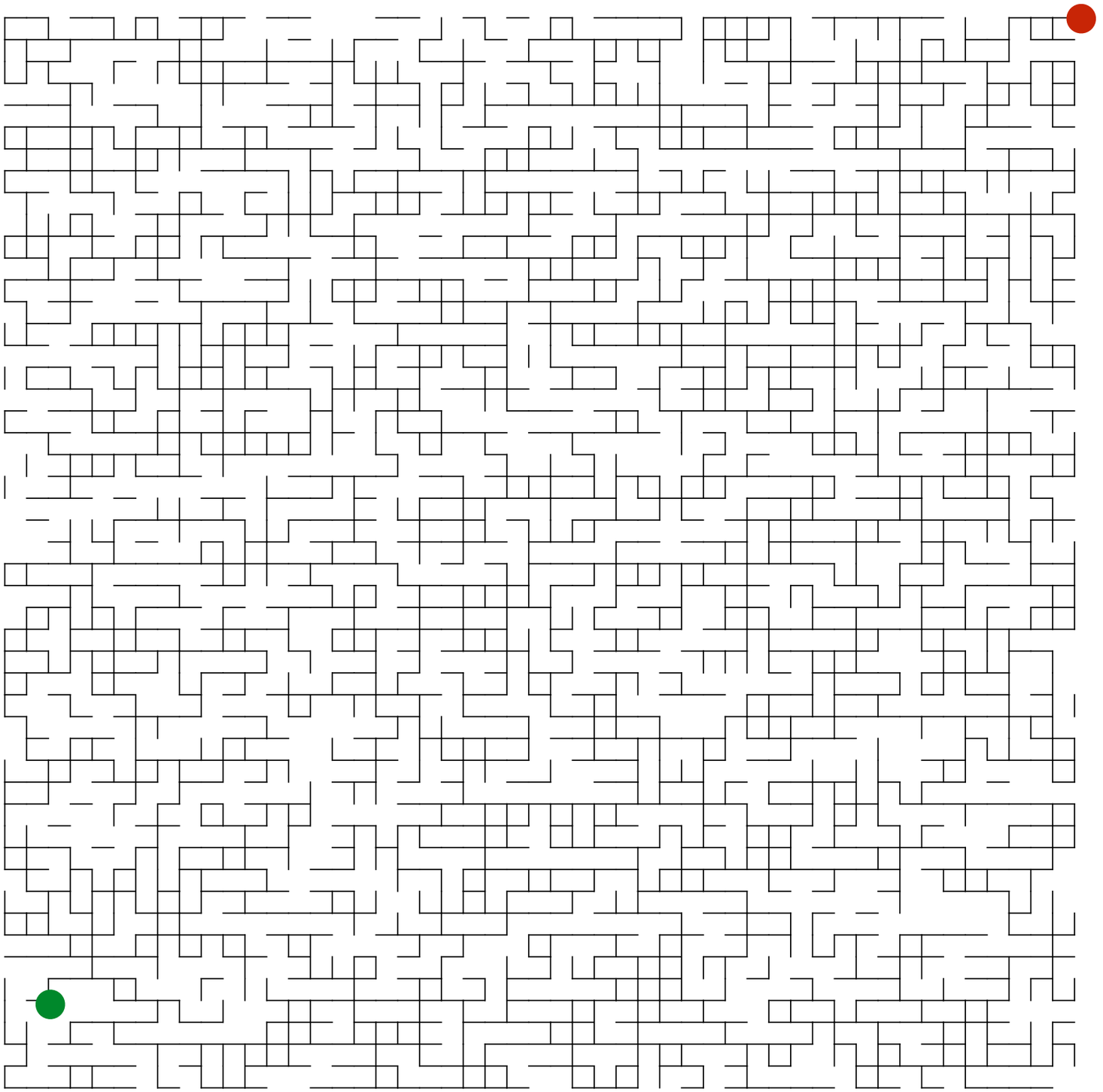
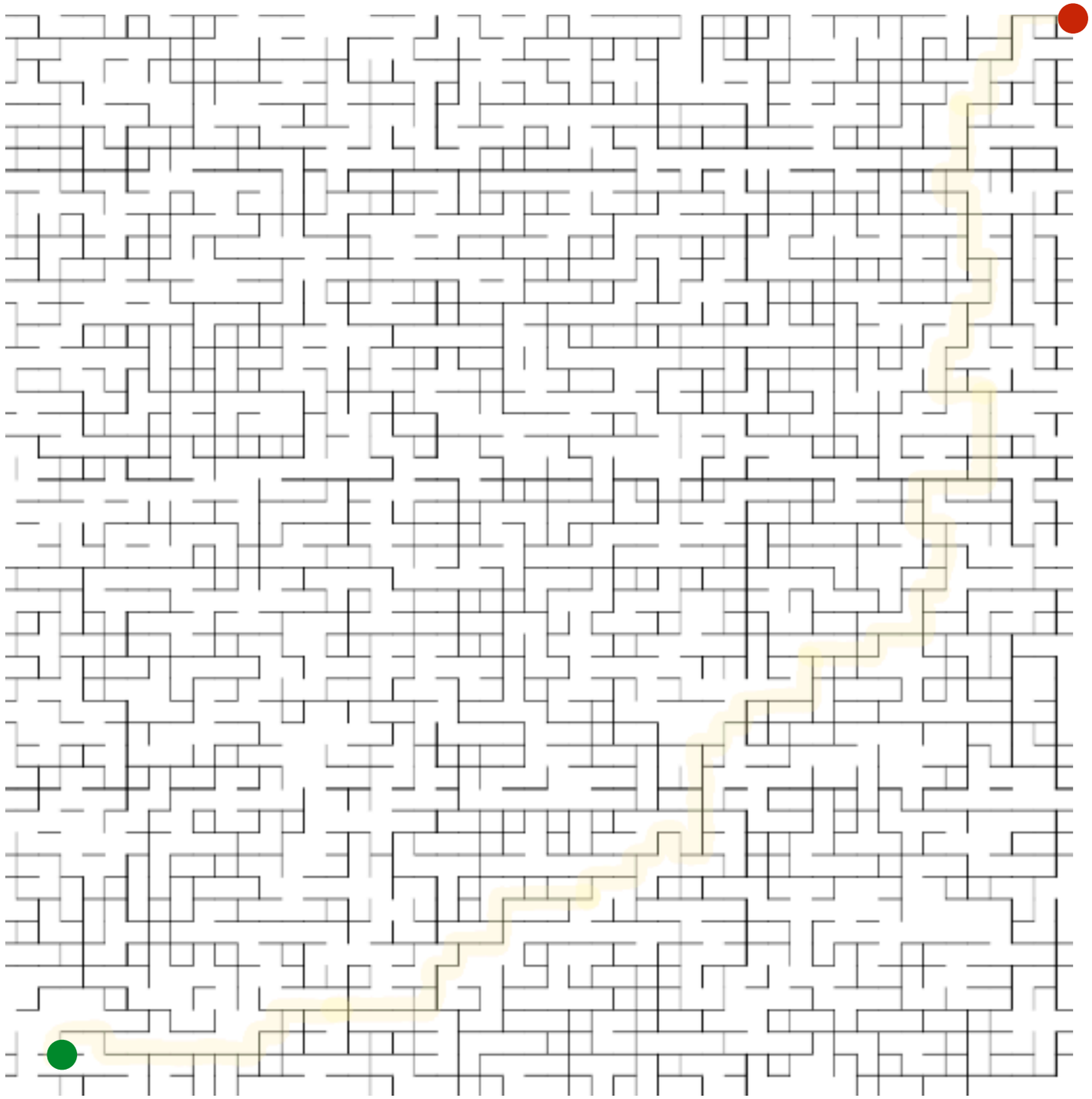
Quantum Walks

Random Walks

If you enter the maze at the green point can you find a path to exit at the red point?



If you enter the maze at the green point can you find a path to exit at the red point?



Maze Problem

How could you solve the maze problem?

Breadth-first search:

Time complexity: $O(n+m)$

Space complexity: $O(n)$

Maze Problem

You can also solve it simply by taking a random walk.

Fact: Let $G = (V, E)$ be an undirected graph with n vertices and m edges, and let $u, v \in V$ be connected.

The expected number of steps for a random walk starting at u to reach v is at most $O(mn)$.

Random walk:

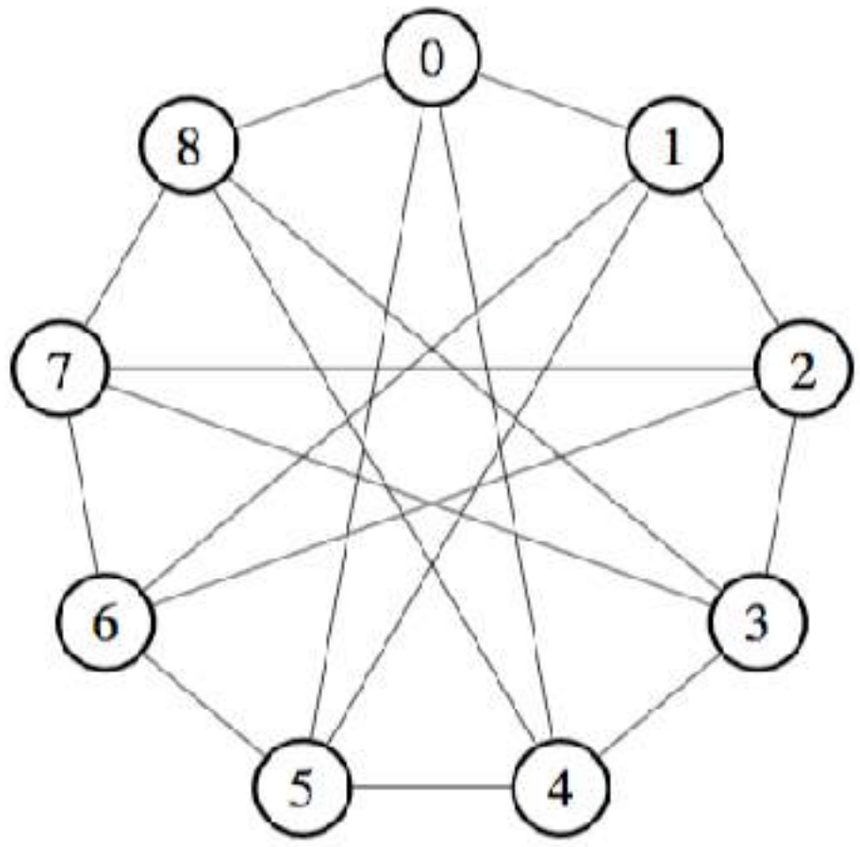
Time complexity: $O(n^3)$

Space complexity: $O(\log n)$

Random Walk

We will restrict to d-regular graphs.

Every vertex has degree d.



Let v be the entrance.

Repeat:

Roll a d-sided die, say outcome is j.

Move to the j^{th} neighbor of v (call it w).

$v \leftarrow w$

Until v is the exit.

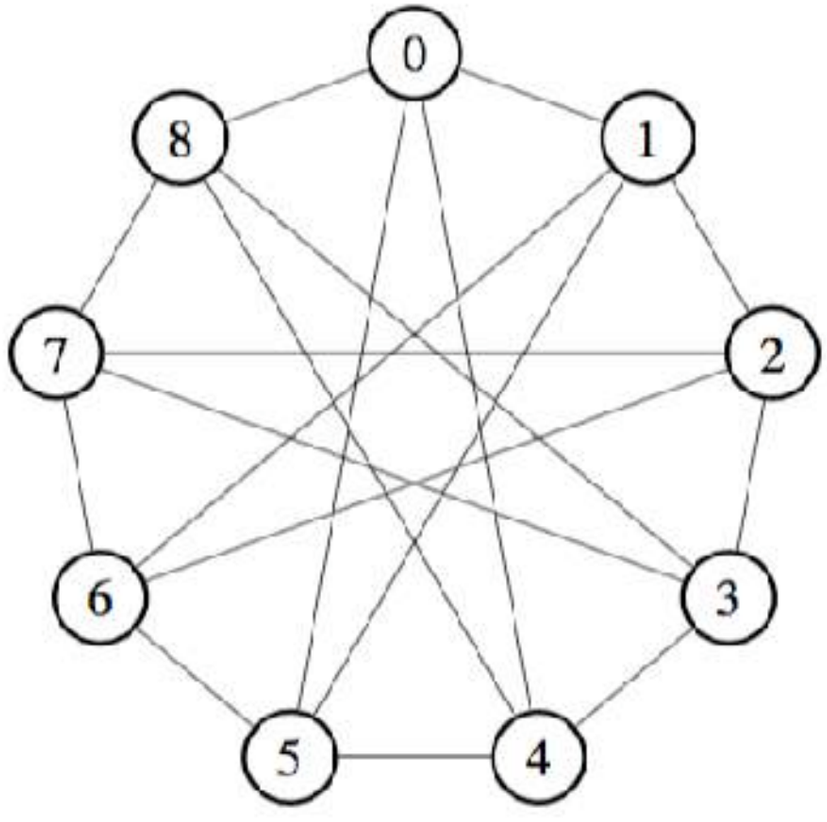
Random Walk

Why does this work?

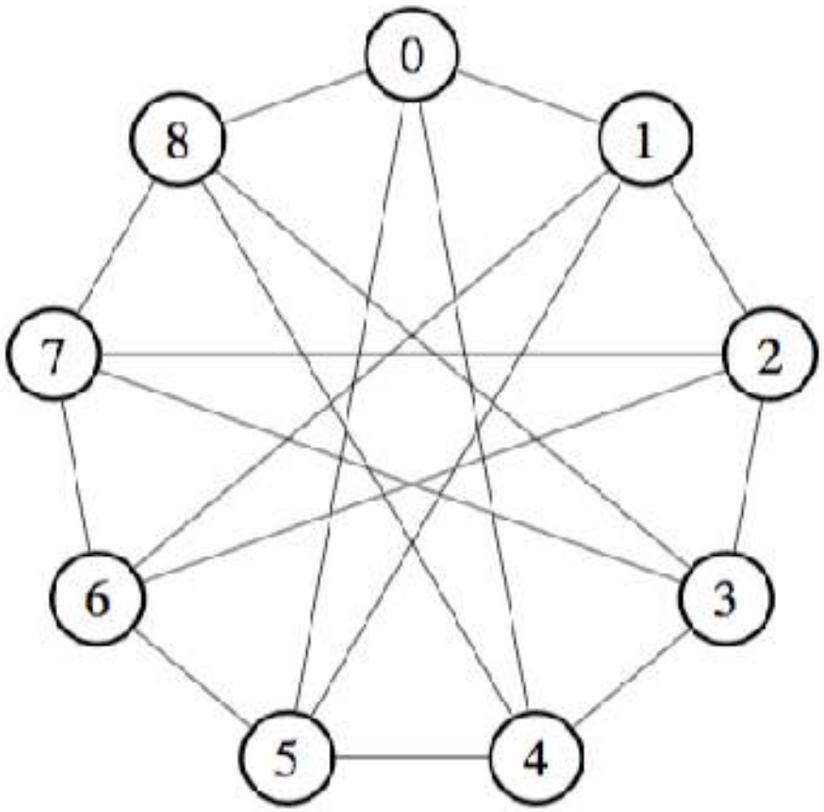
Let $p_t(a)$ be the probability we are at node a after t steps of the walk.

$$p_0(0) = 1$$

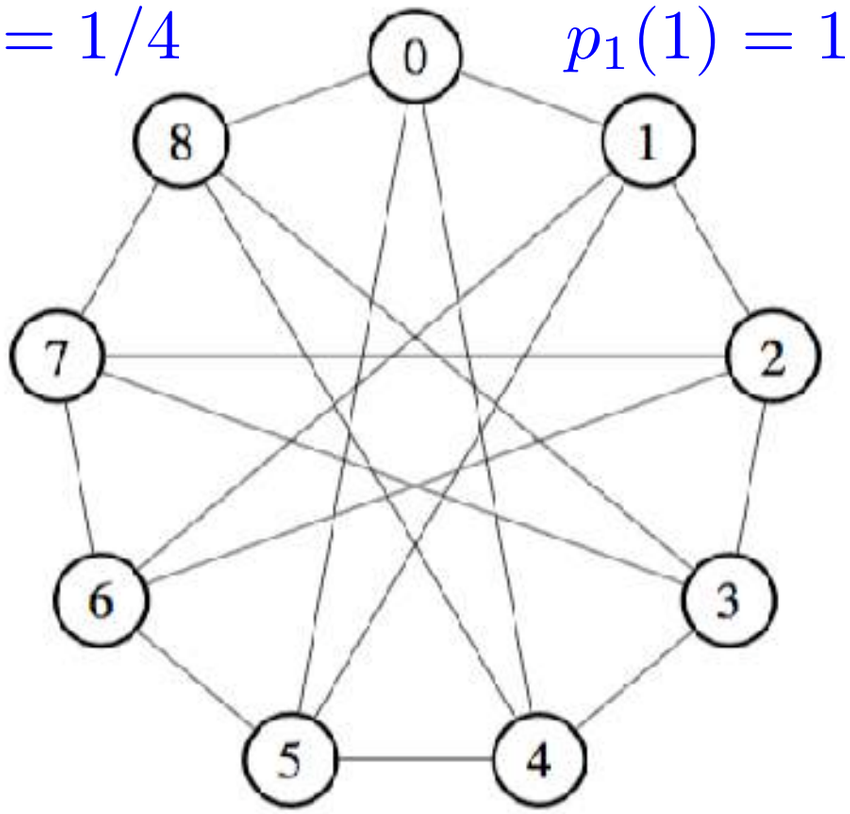
Say we start at vertex 0.



$p_0(0) = 1$



$p_1(8) = 1/4$ $p_1(1) = 1/4$



$p_1(5) = 1/4$ $p_1(4) = 1/4$

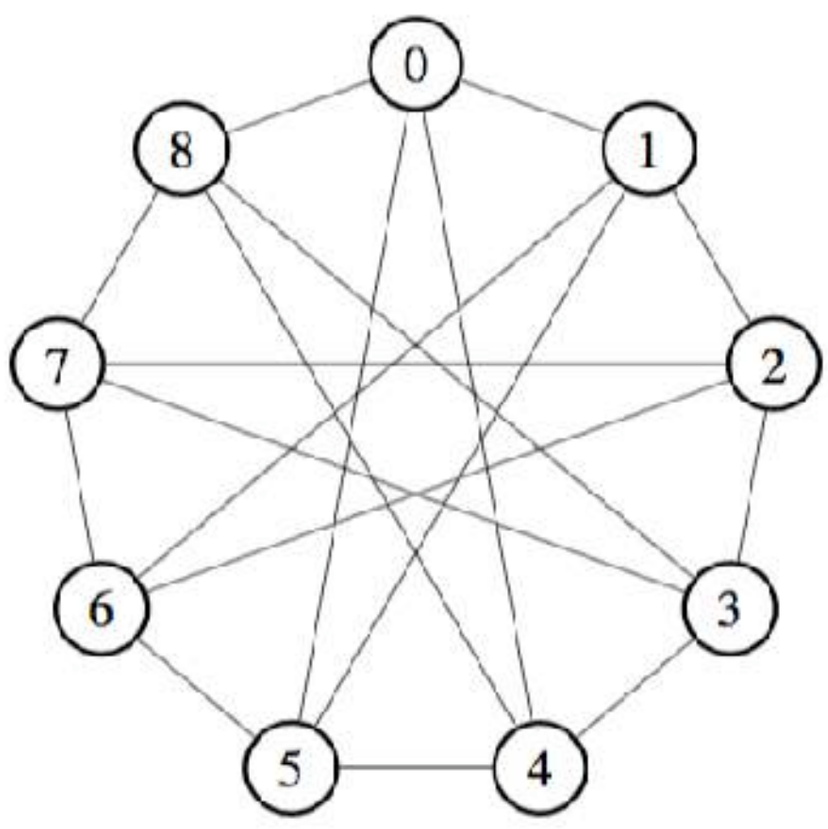
$$p_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \longrightarrow \frac{1}{4} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = p_1$$

Adjacency Matrix

We can phrase this action in terms of the (normalized) adjacency matrix A of the d-regular graph $G = (V, E)$.

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{1}{4}\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Adjacency Matrix

We can phrase this action in terms of the (normalized) adjacency matrix A of the d-regular graph $G = (V, E)$.

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

For any standard basis vector e_i we have that Ae_i is the probability distribution after one step of a walk starting at vertex i.

By linearity, $p_{t+1} = Ap_t$.

Adjacency Matrix

We can phrase this action in terms of the (normalized) adjacency matrix A of the d-regular graph $G = (V, E)$.

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

After t steps of a random walk starting at vertex a

$$p_t = A^t e_a$$

We want to understand this distribution.

Adjacency Matrix

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

- Every row sums to one. This means that for u the uniform distribution $u = 1/n$ we have

$$Au = u$$

The uniform distribution is a stationary distribution.

Once we reach the stationary distribution, as we keep walking the distribution over vertices does not change.

Adjacency Matrix

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

- This matrix is symmetric. It has an orthonormal basis of eigenvectors $v_1, \ldots, v_n$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

We already know $1/\sqrt{n}$ is an eigenvector with eigenvalue 1.

The sum of the entries of any other eigenvector will be zero.

Adjacency Matrix

$$A(a,b) = \begin{cases} \frac{1}{d} & \text{if } \{a,b\} \in E \\ 0 & \text{otherwise} \end{cases}$$

- This matrix is non-negative. By the Perron-Frobenius theorem all eigenvalues have magnitude at most 1.

If the graph is connected then $\lambda_2 < 1$.

If the graph is not bipartite then $\lambda_n > -1$.

From now on we will assume this is the case.

Convergence

With this assumption then $v_1 = \mathbf{1}/\sqrt{n}, \lambda_1 = 1$ and $|\lambda_j| \leq 1 - \delta$ for $j = 2, \ldots, n$ and some $\delta > 0$.

δ is called the spectral gap of the graph.

Let $p_0 = \sum_{i=1}^{n} \alpha_i v_i$ be the starting distribution.

You can think of this as e_a , but the proof works in general.

We know what α_1 is:

$$1 = \mathbf{1}^T p_0 = \sum_{i=1}^{n} \alpha_i \mathbf{1}^T v_i = \alpha_i \mathbf{1}^T v_1 = \alpha_1 \sqrt{n}$$

This means $\alpha_1 v_1 = u$ is the uniform distribution.

$$\begin{aligned} A^t p_0 &= \sum_{i=1}^{n} \alpha_i A^t v_i \\ &= \sum_{i=1}^{n} \alpha_i \lambda_i^t v_i \\ &= u + \sum_{i=2}^{n} \alpha_i \lambda_i^t v_i \end{aligned}$$

Now let's look at the deviation of $A^t p_0$ from the uniform distribution in ℓ_2 distance.

$$A^t p_0 = u + \sum_{i=2}^{n} \alpha_i \lambda_i^t v_i$$

$$\begin{aligned}
\|A^t p_0 - u\|^2 &= \|\sum_{i=2}^{n} \alpha_i \lambda_i^t v_i\|^2 \\
&= \sum_{i=2}^{n} \alpha_i^2 \lambda_i^{2t} \|v_i\|^2 \\
&\leq (1-\delta)^{2t} \sum_{i=2}^{n} \alpha_i^2 \|v_i\|^2 \\
&\leq (1-\delta)^{2t}
\end{aligned}$$

Because $\sum_{i=1}^{n} \alpha_i^2 \|v_i\|^2 = \|p_0\|^2 \leq 1$.

Distance from uniform

The deviation of $A^t p_0$ from the uniform distribution in ℓ_2 distance is at most

$$\|A^t p_0 - u\| \leq (1-\delta)^t$$

Taking $t = \ln(1/\eta)/\delta$ ensures $\|A^t - p_0\| \leq \eta$.

After this many steps the distribution will be close to uniform.

More generally you can use this idea to compute the principal eigenvector of a matrix (power method).

Question

What is the spectral gap of the complete graph on n vertices?

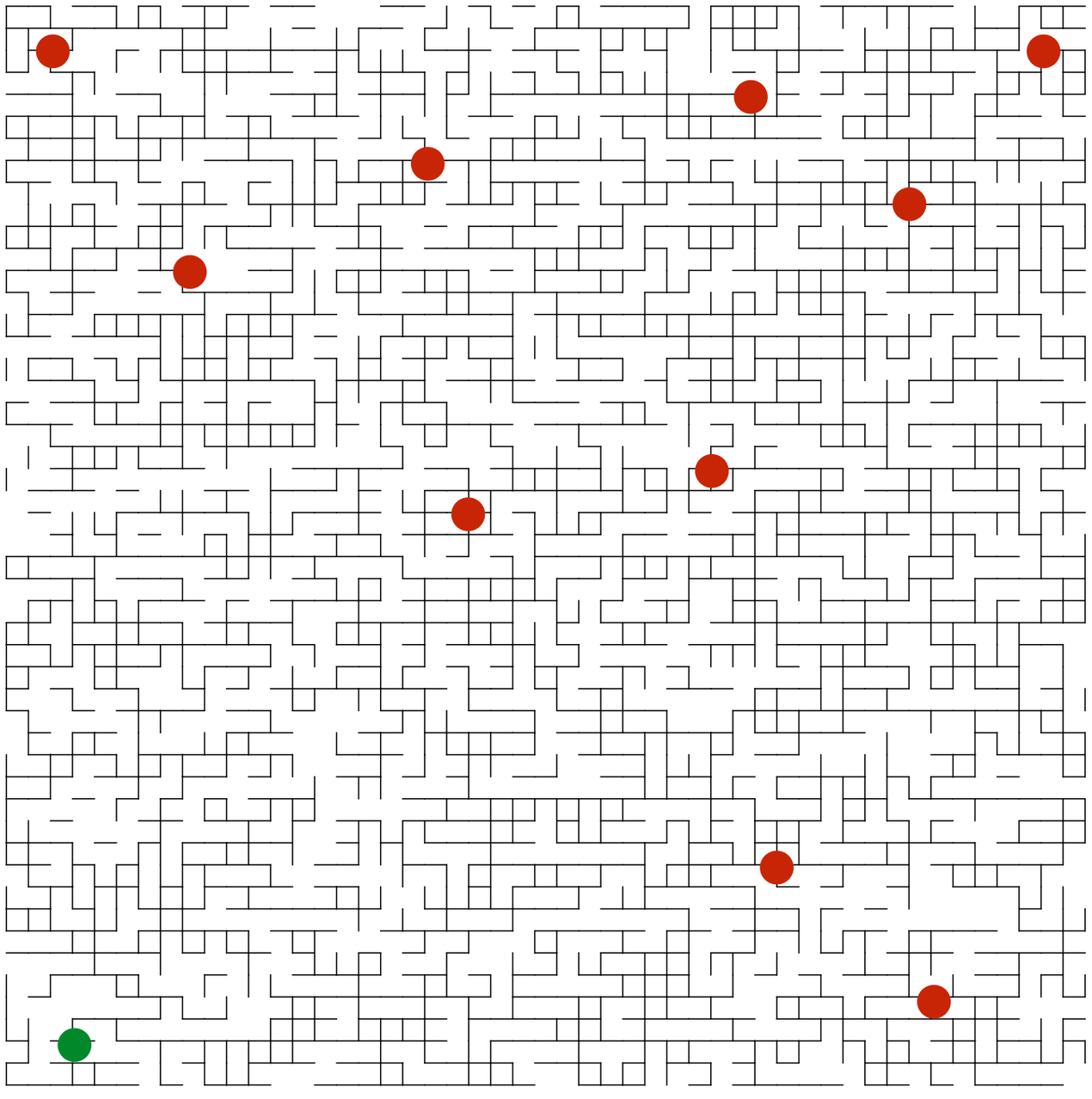
General Graph Search

Search via Random Walk

Now we consider a general scenario.

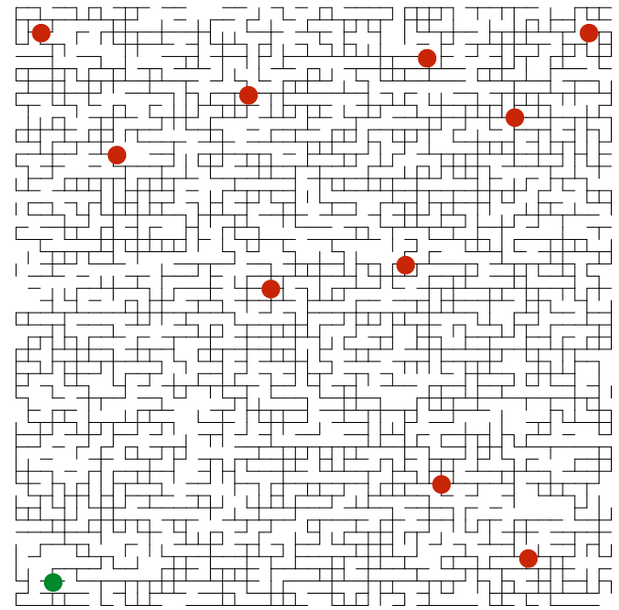
An ε-fraction of vertices are marked.

What is the cost of finding a marked vertex?



Search via Random Walk

Idea: Say the spectral gap of the graph is δ.



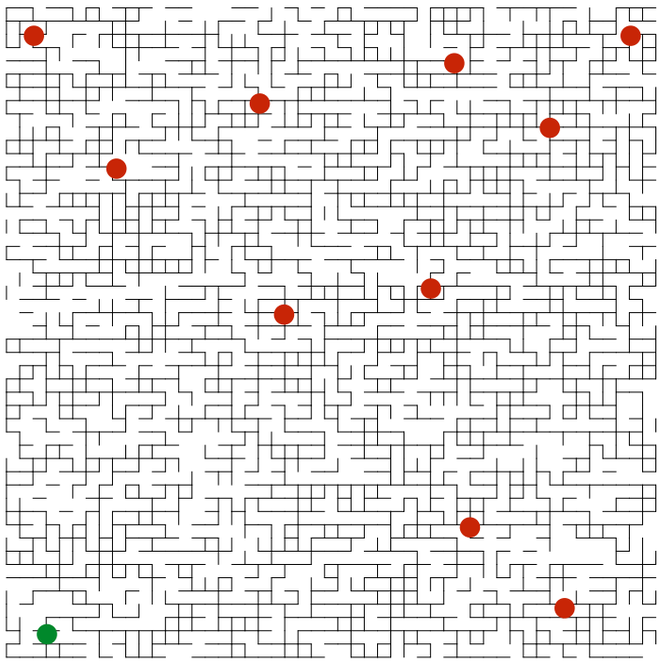
From any starting vertex, after $O(1/\delta)$ steps we will be close to the uniform distribution.

Under the uniform distribution, the probability of being at a marked vertex is ε.

Repeating this process $O(1/\varepsilon)$ times we find a marked vertex with constant probability.

Search via Random Walk

For our applications the graph is known.



But each vertex will have some associated data that is unknown.

Whether or not the vertex is marked will depend on this data.

We will incur costs for learning the data associated to a vertex, and checking if a vertex is marked.

Search via Random Walk

We formulate the cost in an abstract way.

Cost could be queries or circuit size.

S Cost to set up data at starting vertex.

U Cost to perform one step of the walk.

C Cost to check if a vertex is marked.

Total Cost: $$\mathsf{S} + \frac{1}{\varepsilon}\left(\frac{1}{\delta}\mathsf{U} + \mathsf{C}\right)$$

Element Distinctness

The input to the element distinctness problem is a string $x \in \{0, \ldots, m-1\}^n$ with $m \geq n$.

We want to determine whether or not all the entries of x are distinct.

This problem is closely related to the collision problem of determining if a function is 2-to-1 or 1-to-1.

Johnson Graph

Let's consider solving element distinctness by a random walk on the Johnson graph $J(n, r)$.

The vertices are all r-element subsets of $\{1, \ldots, n\}$.

Two vertices are connected by an edge iff their intersection size is $r - 1$.

For $r \leq n/2$ the eigenvalue gap is known to be $\Theta(1/r)$.

Johnson Graph

For a vertex $T \subseteq \{1, \ldots, n\}$ the associated data is $\{x_i : i \in T\}$.

The vertex T is marked iff $\exists i, j \in T$ with $x_i = x_j$.

Let's look at the costs in terms of queries to x.

If we know the data associated to a vertex, the checking cost $\mathrm{C} = 0$.

Johnson Graph

The setup cost is to learn the data associated to the starting vertex.

The setup cost $\mathsf{S} = r$ starting at a fixed vertex.

From a vertex T we can learn the data of a neighboring vertex T' with just one query.

The update cost $\mathsf{U} = 1$.

Overall Cost

The probability a vertex is marked $\varepsilon = \Omega((r/n)^2)$.

Total Cost: $$\mathsf{S} + \frac{1}{\varepsilon}\left(\frac{1}{\delta}\mathsf{U} + \mathsf{C}\right)$$

$$= r + \frac{n^2}{r^2} r$$

This is minimized by taking $r = n$ just giving an $O(n)$ algorithm.

In fact the randomized query complexity is $\Theta(n)$.

Question

Say you are given an algorithm for the element distinctness problem. How can you use this to solve the collision problem?

Quantum Walks

Quantum Search on a Graph

We start at the end, with the result we will sketch.

Theorem [MNRS11]: Let $G = (V, E)$ be a d-regular graph with spectral gap δ and where an ε-fraction of vertices are marked.

There is a quantum algorithm to find a marked vertex with cost

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}} \left(\frac{1}{\sqrt{\delta}} \mathsf{U} + \mathsf{C} \right)$$

S setup cost, U update cost, C checking cost.

Element Distinctness

Let's see an application of this to element distinctness.

I haven't formally defined $\mathsf{S}, \mathsf{U}, \mathsf{C}$ in the quantum case, but for this example they are all the same as the classical.

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right) = r + \frac{n}{r}\sqrt{r}$$

Now the minimum is obtained with $r = n^{2/3}$.

The quantum query complexity of element distinctness is $\Theta(n^{2/3})$.

quant-ph/0311001

High Level Picture

The MNRS theorem is based on Grover's algorithm.

We will alternate reflections about the "bad" state with reflections about the "uniform" state.

The good state has overlap ε with the uniform state.

We do $O(1/\sqrt{\varepsilon})$ rounds of Grover to get constant overlap with the good state.

High Level Picture

The complexity of reflecting about the bad state is C.

The interesting part is reflecting about the uniform state, and this is where the quantum walk comes in.

Reflecting about the uniform state will take $O(1/\sqrt{\delta})$ applications of the quantum walk operator.

The cost of this reflection will be $U/\sqrt{\delta}$.

Total Cost

There is a quantum algorithm to find a marked vertex with cost

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$

- S: set up $|U\rangle$
- $\frac{1}{\sqrt{\varepsilon}}$: # rounds of Grover iterate
- $\frac{1}{\sqrt{\delta}}$: reflection about $|U\rangle$
- C: reflection about $|\psi_{\mathrm{bad}}\rangle$

Here $|U\rangle$ is the "uniform" state, $|\psi_{\mathrm{bad}}\rangle$ is the "bad" state.

Quantum Walks

It is not so obvious how to define a quantum analog of a random walk on a d-regular graph $G = (V, E)$.

First attempt: have a map that sends a vertex to a uniform superposition of its neighbors.

$$|j\rangle \mapsto \frac{1}{\sqrt{d}} \sum_{\substack{k \\ \{j,k\} \in E}} |k\rangle$$

What is wrong with this?

Quantum Walks

We will instead think about walking on the edges of the graph.

Our states will have two registers, each labeled by a vertex.

Interpret being in state $|j\rangle|k\rangle$ to mean we are currently at vertex j and came from vertex k.

With this choice we can now define the "good" and "uniform" states for our Grover setup.

Grover Setup

Let M be the set of marked vertices and let $\frac{|M|}{n} = \varepsilon < \frac{1}{2}$.

For a vertex j let $|N_j\rangle = \frac{1}{\sqrt{d}} \sum_{\substack{k \\ \{k,j\} \in E}} |k\rangle$.

The "good" state is $|\psi_{\text{good}}\rangle = \frac{1}{\sqrt{|M|}} \sum_{j \in M} |j\rangle |N_j\rangle$.

The "bad" state is $|\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{n - |M|}} \sum_{j \notin M} |j\rangle |N_j\rangle$.

Grover Setup

The uniform state is

$$|U\rangle = \frac{1}{\sqrt{dn}} \sum_{j \in V} \sum_{\substack{k \\ \{k,j\} \in E}} |j\rangle |k\rangle$$

$$= \frac{1}{\sqrt{n}} \sum_{j \in V} |j\rangle |N_j\rangle$$

Just as in the standard Grover setting,

$$|U\rangle = \sin(\theta)|\psi_{\text{good}}\rangle + \cos(\theta)|\psi_{\text{bad}}\rangle$$

where $\theta = \arcsin(\sqrt{\epsilon})$.

Grover Setup

With this setup we can emulate Grover.

1) Start in the state $|U\rangle$.

2) Repeat $\left\lfloor \frac{\pi}{4\theta} \right\rfloor = O(1/\sqrt{\varepsilon})$ many times

- Reflect about $|\psi_{\mathrm{bad}}\rangle$.
- Reflect about $|U\rangle$.

3) Measure first register and see if vertex is marked.

We start in the state $|U\rangle = \sin(\theta)|\psi_{\text{good}}\rangle + \cos(\theta)|\psi_{\text{bad}}\rangle$.

After k applications of the repeat loop we are in the state

$$\sin((2k+1)\theta)|\psi_{\text{good}}\rangle + \cos((2k+1)\theta)|\psi_{\text{bad}}\rangle$$

As we have argued before, the choice

$$k = \left\lfloor \frac{\pi}{4\theta} \right\rfloor = O(1/\sqrt{\varepsilon})$$

makes the success probability at least $1 - \varepsilon$.

Implementing the reflections

- Reflect about $|\psi_{\text{bad}}\rangle$.

  To do this it suffices to be able to check if a vertex is marked or not.

  We can perform this with cost C.

- Reflect about $|U\rangle$.

  This is the interesting step where the quantum walk comes in.

  We explain this next.

Defining the quantum walk

Let's think about how we could define a classical walk on the edges of a d-regular graph $G = (V, E)$.

We could do this by combining two operations:

• Put a random neighbor of the vertex in the first register in the second register:

$$|j\rangle|k\rangle \to |j\rangle|\ell\rangle \qquad \{j, \ell\} \in E$$

• Swap the two registers

$$|j\rangle|\ell\rangle \to |\ell\rangle|j\rangle$$

Walking on edges

For developing a quantum analog, it is more convenient to think of two steps of this random walk.

- Put a random neighbor of the vertex in the first register in the second register:

$$|j\rangle|k\rangle \to |j\rangle|\ell\rangle \qquad \{j,\ell\} \in E$$

- Put a random neighbor of the vertex in the second register in the first register:

$$|j\rangle|\ell\rangle \to |k\rangle|\ell\rangle \qquad \{\ell,k\} \in E$$

Quantum Walk

We develop a quantum analog of this process.

Recall $|N_j\rangle = \frac{1}{\sqrt{d}} \sum_{\substack{k \\ \{k,j\}\in E}} |k\rangle$ for a vertex j .

Let $W_1 = \sum_{j\in V} |j\rangle\langle j| \otimes (2|N_j\rangle\langle N_j| - \mathbf{I})$

$$W_2 = \sum_{j\in V} (2|N_j\rangle\langle N_j| - \mathbf{I}) \otimes |j\rangle\langle j|$$

The quantum walk unitary is $W = W_2 W_1$.

Spectrum of W

Let $\mathcal{Z} = \mathrm{span}\left\{\cup_{j\in V}|j\rangle|N_j\rangle, \cup_{j\in V}|N_j\rangle|j\rangle\right\}$.

We can relate the eigenvalues of eigenvectors of W in $\mathcal{Z}$ to the spectrum of A the normalized adjacency matrix of G.

The state $|U\rangle = \frac{1}{\sqrt{n}}\sum_{j\in V}|j\rangle|N_j\rangle$ is fixed by both W_1 and W_2.

$|U\rangle$ is an eigenvector of W with eigenvalue 1.

If G is connected and not bipartite this is the only eigenvector with eigenvalue 1 in $\mathcal{Z}$.

Spectrum of W

Let $\lambda_1, \ldots, \lambda_\ell$ be those eigenvalues of A with $|\lambda_j| \in (0, 1)$.

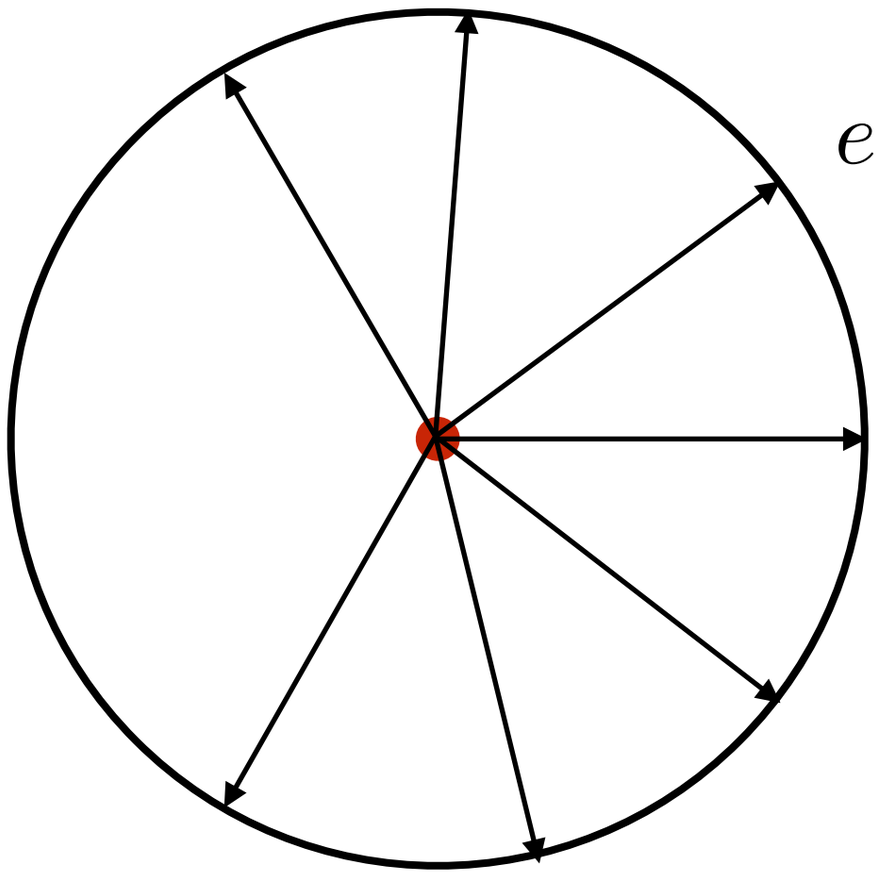
On $\mathcal{Z}$, the eigenvalues of W with nonzero imaginary part are

$$e^{-2\mathrm{i}\theta_1}, e^{2\mathrm{i}\theta_1}, \ldots, e^{-2\mathrm{i}\theta_\ell}, e^{2\mathrm{i}\theta_\ell}$$

where $\cos(\theta_j) = |\lambda_j|$.

All remaining eigenvalues of W on $\mathcal{Z}$ have value -1 .

Spectrum of W



$e^{2\mathrm{i}\arccos(1-\delta)}$

eigenvalue 1 with eigenvector $|U\rangle$.

$1-\delta = \cos(\theta) \geq 1-\theta^2/2$

All other eigenvalues have angle at least $\sqrt{2\delta}$ because the spectral gap of A is δ.

Reflection about uniform

All other eigenvalues have angle at least $\sqrt{2\delta}$ because the spectral gap of A is δ.

We can use phase estimation to implement the reflection about $|U\rangle$.

Idea: Given an eigenvector $|v\rangle$ of W, estimate its phase to within $\sqrt{\delta}/2$.

If magnitude of estimate is bigger than $\sqrt{\delta}/2$ then multiply $|v\rangle$ by -1, otherwise do nothing.

Reflection about uniform

Idea: Given an eigenvector $|v\rangle$ of W, estimate its phase to within $\sqrt{\delta}/2$.

If magnitude of estimate is bigger than $\sqrt{\delta}/2$ then multiply $|v\rangle$ by -1, otherwise do nothing.

Obtaining this estimate can be done with $O(1/\sqrt{\delta})$ many applications of W.

The update cost U is the cost of implementing W.

Update Cost

To implement W it suffices to implement W_1 and W_2.

$$W_1 = \sum_{j \in V} |j\rangle\langle j| \otimes (2|N_j\rangle\langle N_j| - \mathbf{I})$$

To implement W_1 it suffices to be able to do the map

$$|j\rangle|0\rangle \mapsto |j\rangle|N_j\rangle$$

and its inverse.

Apply the inverse of this map, multiply by -1 if the second register is not $|0\rangle$, apply this map.

Re-deriving Grover

We saw that Grover's algorithm can find a 1 in a string $x \in \{0,1\}^n$ with εn many ones after $O(1/\sqrt{\varepsilon})$ many queries (where ε is known).

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$

Can you use this formula to re-derive this result?

What graph do you take to do the walk on?

Further Applications

Triangle Finding: Giving oracle access to the adjacency matrix of a graph, determine if the graph has a triangle.

An early application of quantum walks gave an algorithm to tell if an n-vertex graph has a triangle with $O(n^{13/10})$ queries.

quant-ph/0310134

The randomized query complexity is $\Theta(n^2)$.

The current best quantum algorithm makes $O(n^{5/4})$ queries.

arXiv:1407.0085