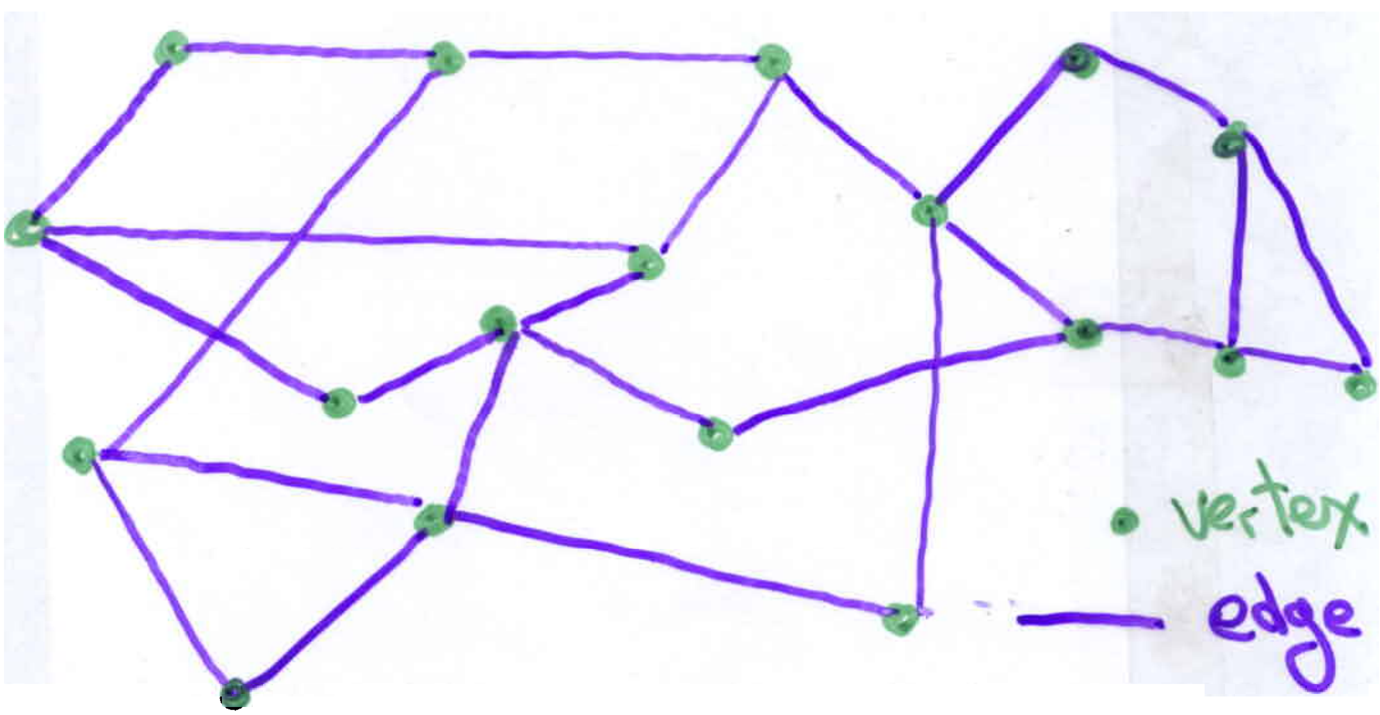
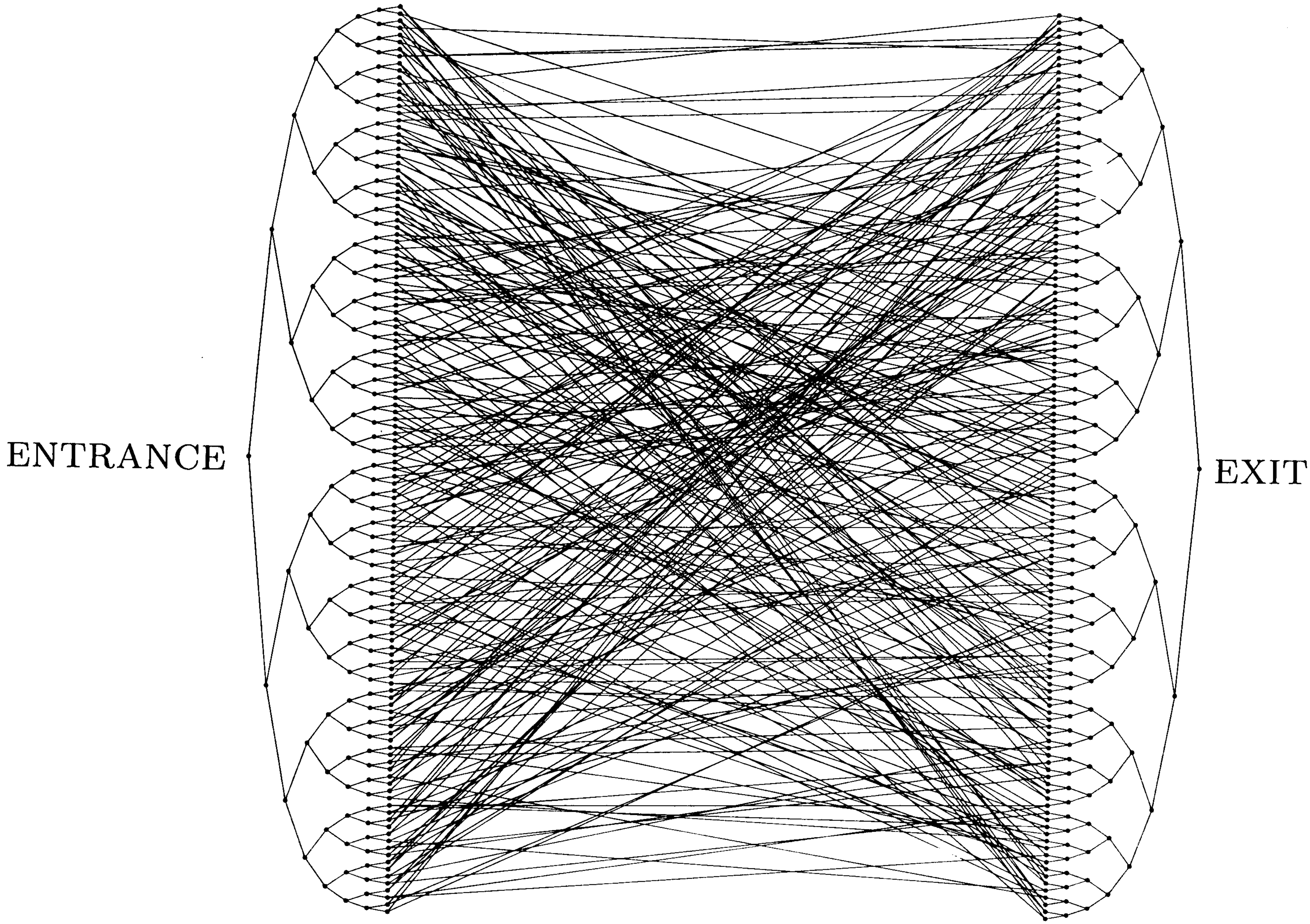
Speedup by Continuous Time Quantum Walk

* outperform classical random walk

* outperform any classical algorithm



2.



Graph G (undirected)

vertices $a \in G$

edges $ab \in G$ $(ab = ba)$

$d(a)$ = # of edges incident on a

Adjacency matrix $A_{ab} = \begin{cases} 1 & ab \in G \\ 0 & \text{otherwise} \end{cases}$

Laplacian $L_{ab} = \begin{cases} -A_{ab} & a \neq b \\ d(a) & a = b \end{cases}$

Classical Random Walk

Continuous time Markov Process

Move to connected vertices with probability per unit time γ

$p_a(t)$ = probability to be at a at time t

$$\boxed{\frac{d p_a(t)}{dt} = -\gamma \sum_b L_{ab}\, p_b(t)}$$

$\boxed{\text{Quantum Walk}}$ quant-ph/9706062

Vector space $\quad \dim = \#$ of vertices of G

Basis $|a\rangle \quad \langle a|b\rangle = \delta_{ab}$

$$\boxed{\text{Define } H \qquad \langle a|H|b\rangle = \gamma L_{ab}}$$

Sch Eq $\rightarrow$ $i\dfrac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$

$$i\frac{d}{dt}\langle a|\psi(t)\rangle = \sum_b \gamma L_{ab}\langle b|\psi(t)\rangle$$

Quantum $\langle a|\psi(t)\rangle = \sum_b \langle a|e^{-iHt}|b\rangle\langle b|\psi(0)\rangle$

classical $\quad p_a(t) = \sum_b \langle a|e^{-Ht}|b\rangle\, p_b(0)$

Suppose $\quad p_b(0) = \langle b|\psi(0)\rangle$

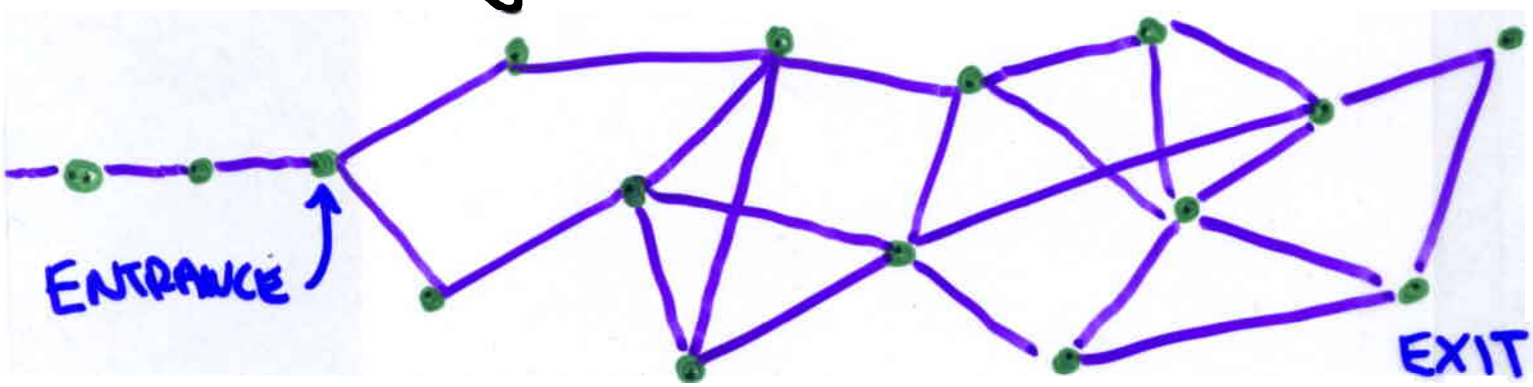
$$\boxed{p_a(t) = \langle a|\psi(-it)\rangle}$$

$$\boxed{p_a(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt' \, \frac{\langle a|\psi(t')\rangle}{t-it'}}$$

Compare Quantum and Classical

Family of Graphs $G_1, G_2, \ldots G_n \ldots$

$n = \log(\#\text{ vertices})$



$t=0$ Start at ENTRANCE

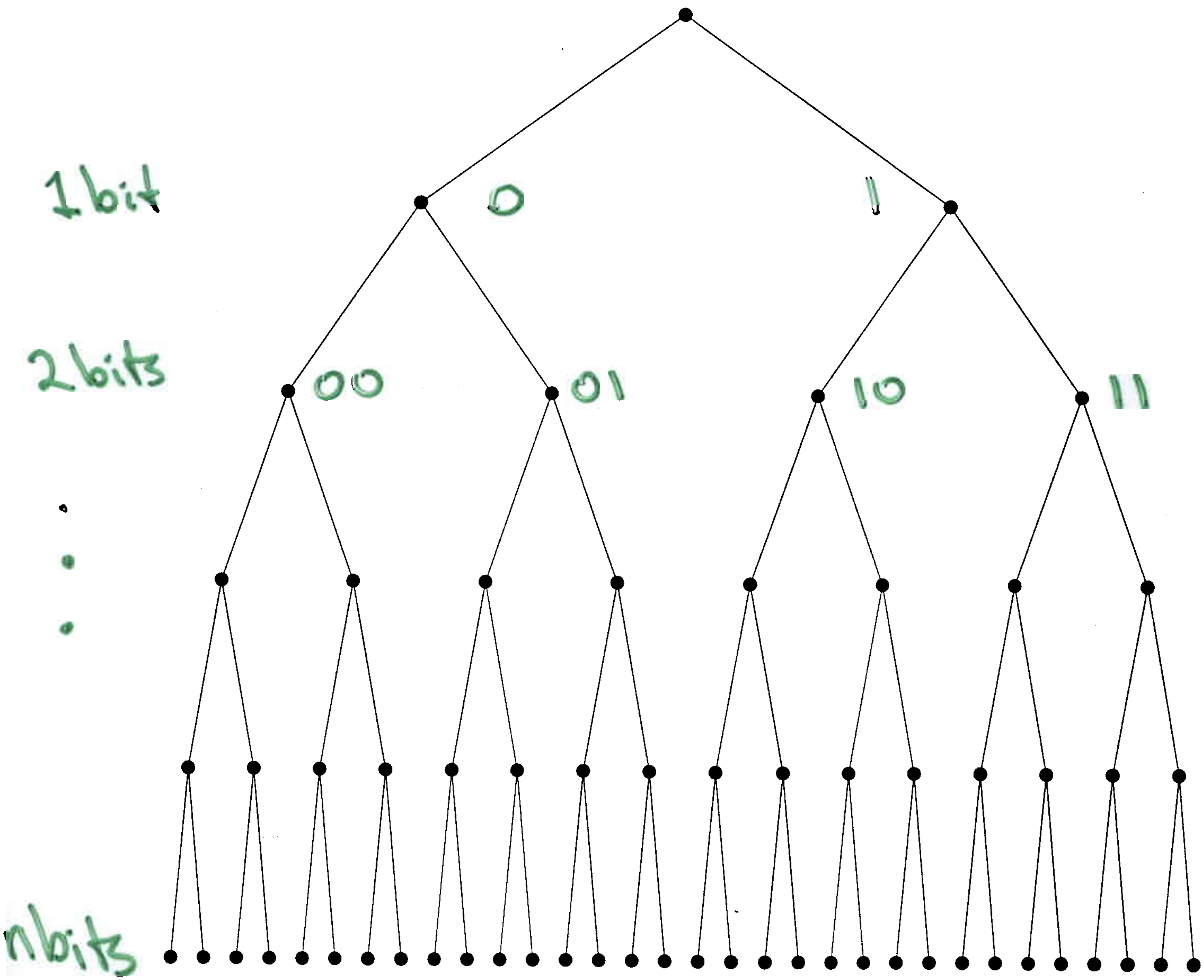
Classically Penetrable : $\exists\ A, B > 0$ such that there is a $t < n^A$ with $P_{EXIT}(t) \geq \frac{1}{n^B}$.

Quantum Penetrable : $\exists\ A', B' > 0$ such that there is a $t < n^{A'}$ with $|\langle EXIT | \psi(t) \rangle|^2 \geq \frac{1}{n^{B'}}$

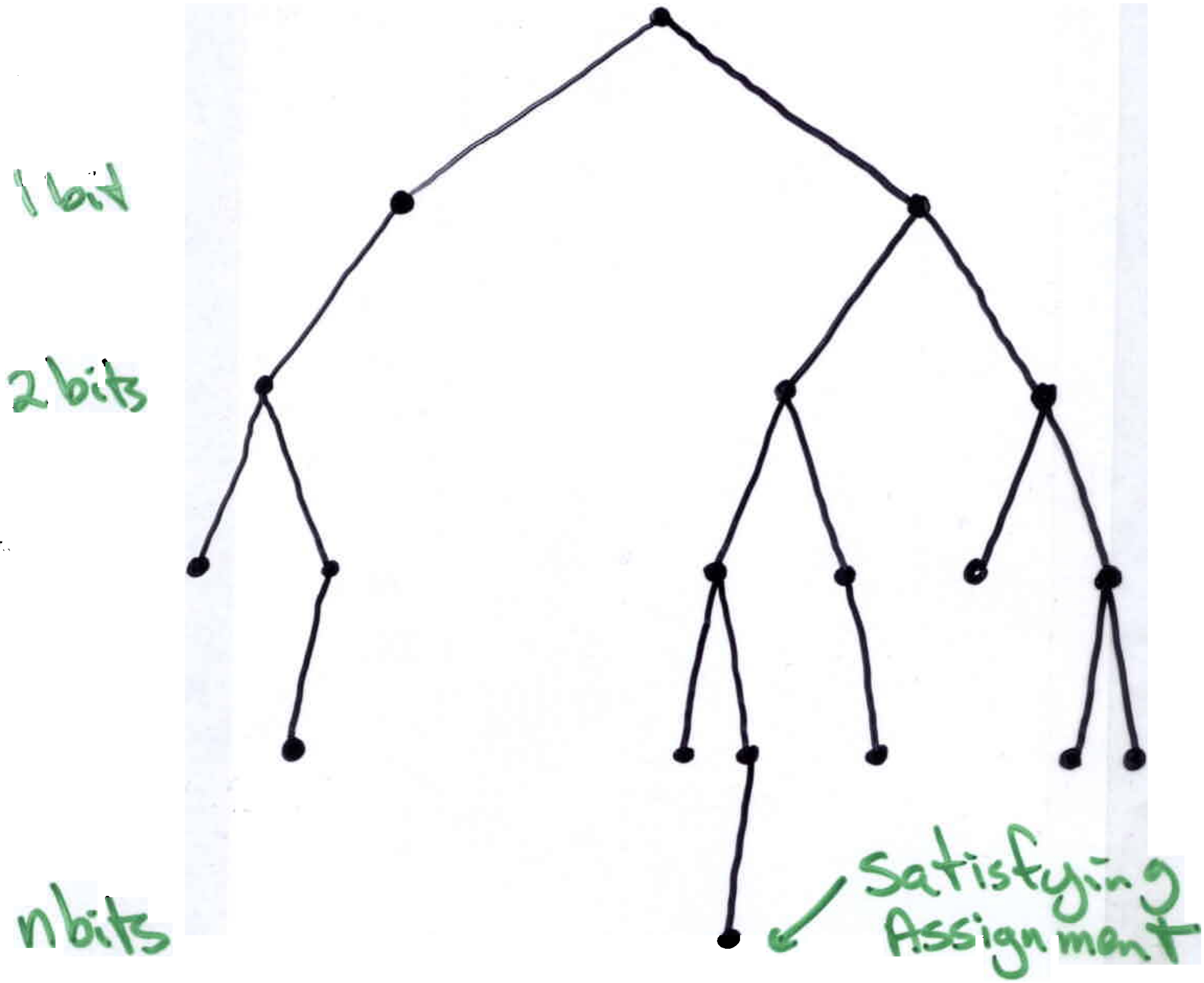
$\boxed{\text{Classically Penetrable} \Longrightarrow \text{Quantum Penetrable}}$

quant-ph 9706062

Possible Algorithmic Use

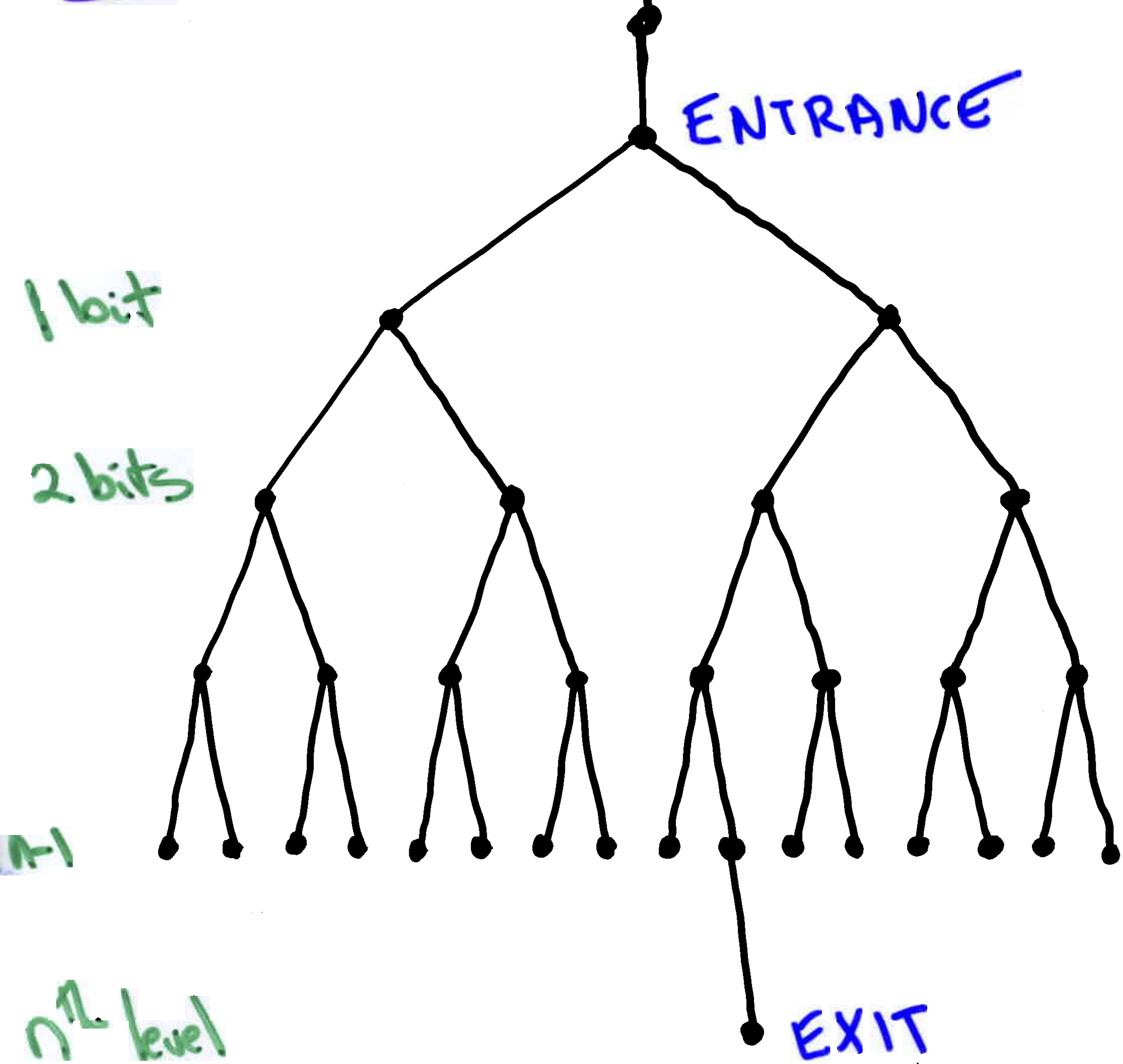


TrimTree According to constraints imposed by a problem



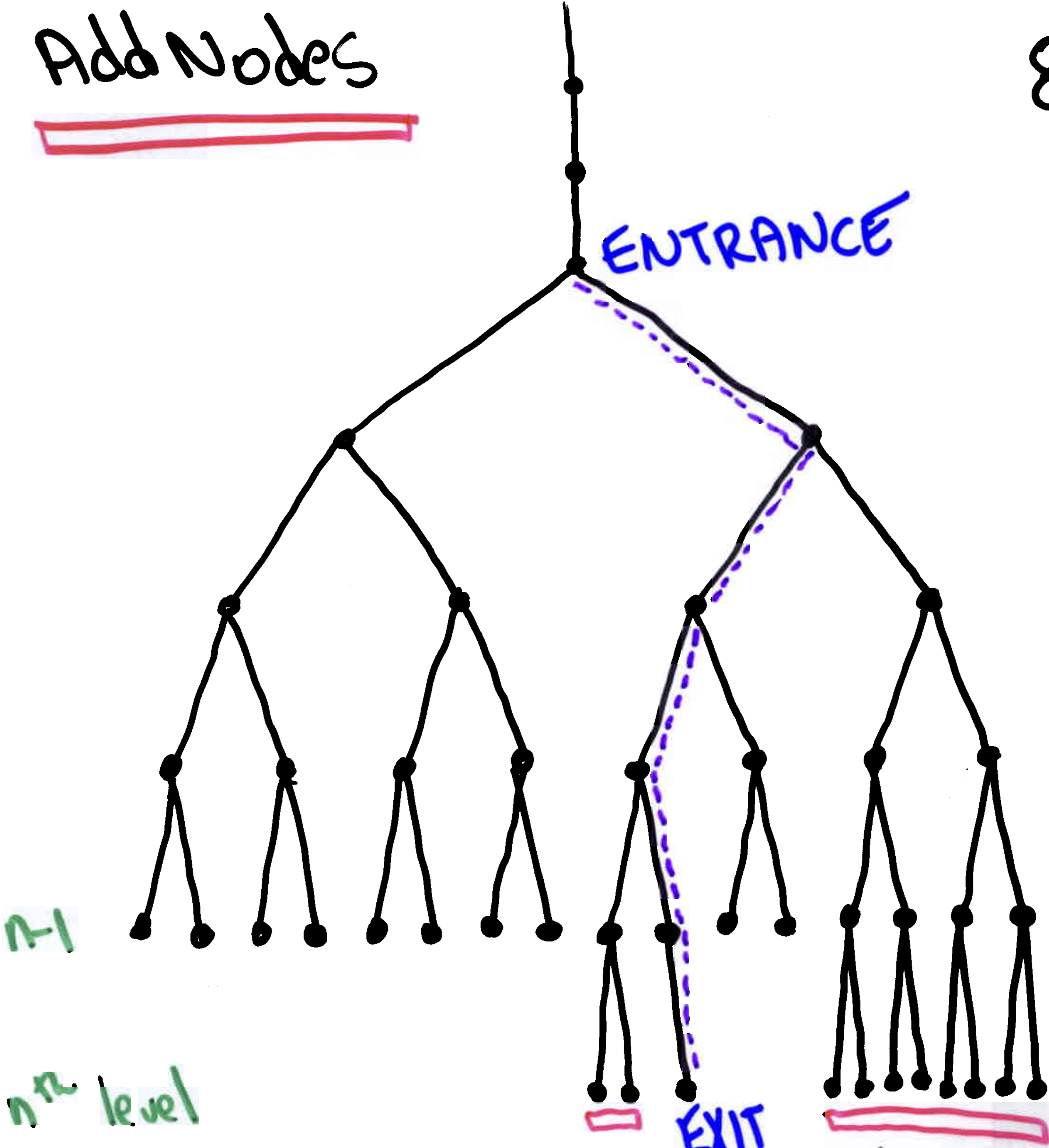
Is there a node at the n^{th} level?

Grover Problem



One constraint at the n^{th} level
Do not penetrate classically or
quantum mechanically

Add Nodes



Classical Probability goes down

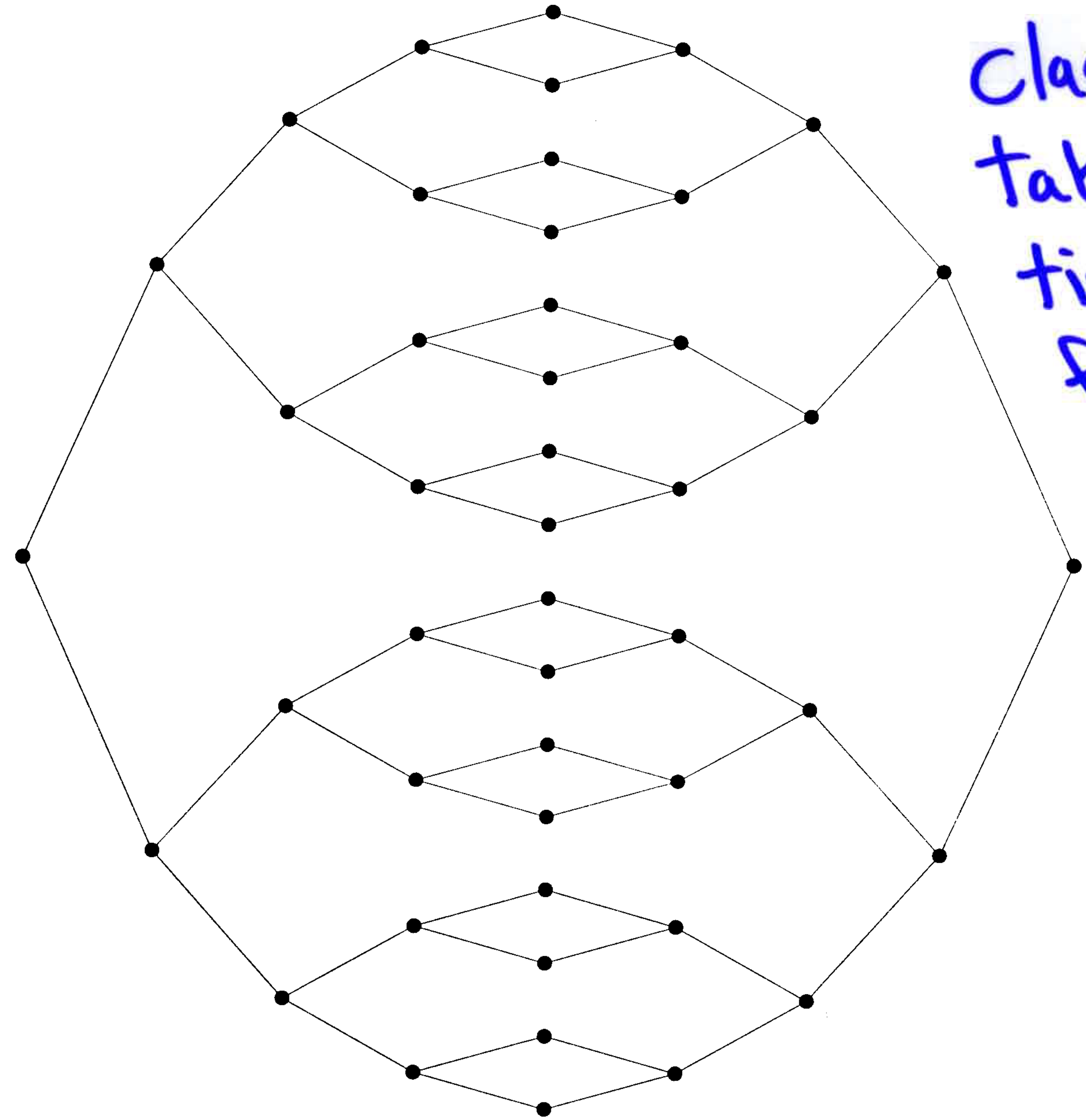
Quantum Mechanically Fly through

Exponential Speed up

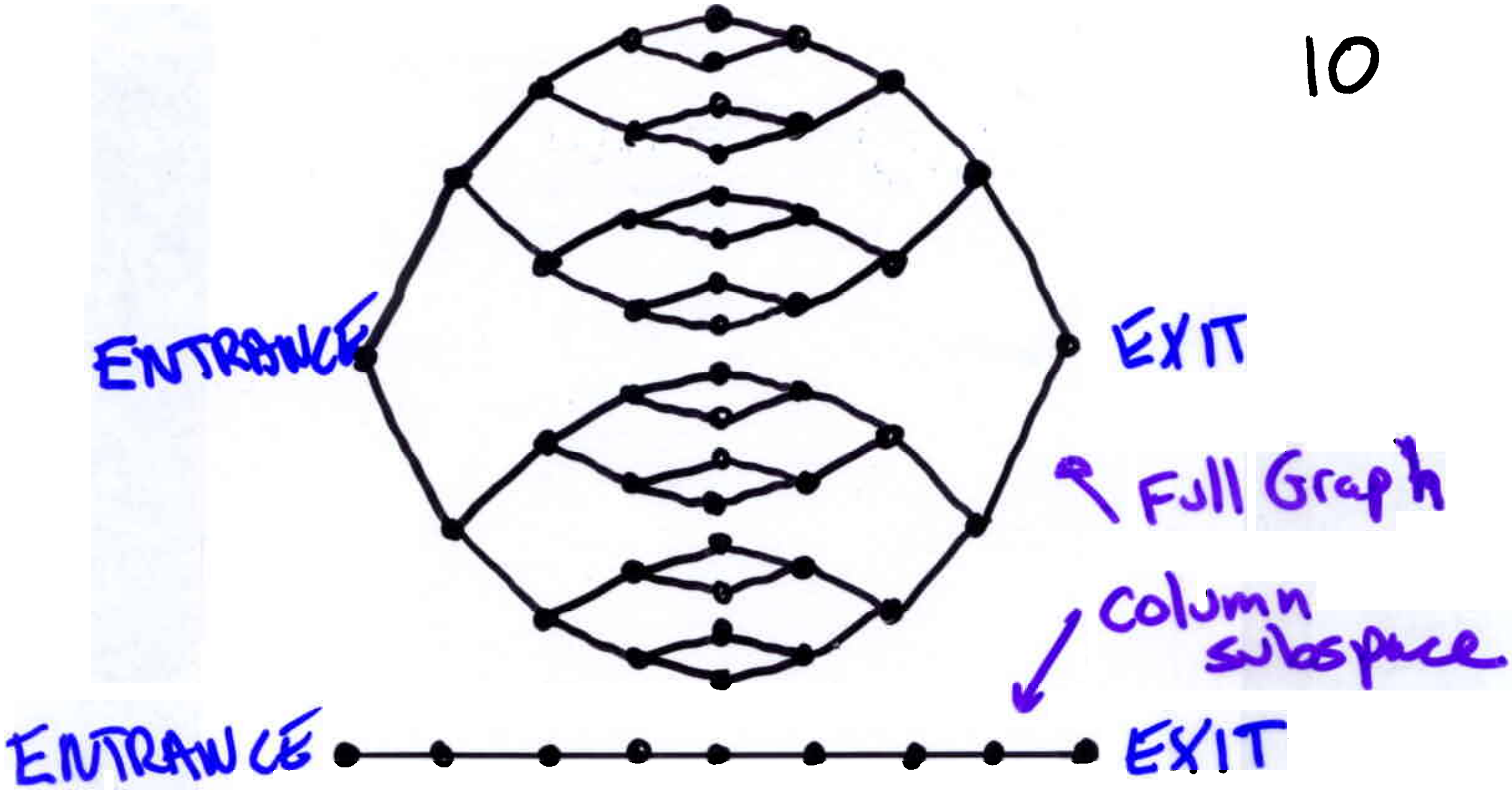
Another Example

$\swarrow 2^n$ nodes

Classical R.W. takes exponential time to go from ENTRANCE to EXIT



A.M. Childs, E. Farhi, S. Gutmann 2001



Let $\langle a|H|b\rangle = \gamma A_{ab}$ Adjacency matrix (diagonal is 0)

$$|\text{col } j\rangle = \frac{1}{\sqrt{N_j}} \sum_{a \in \text{column } j} |a\rangle$$

The column subspace is invarient !!

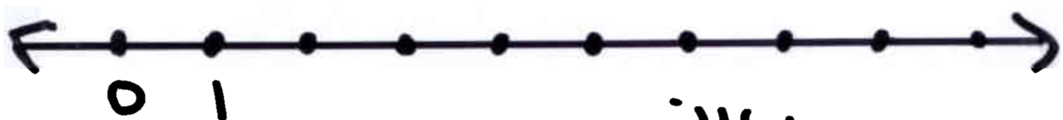
$$\langle \text{col } j|H|\text{col } j\pm1\rangle = \sqrt{2}\,\gamma$$

Quantum walk is reduced to a walk on a line !

$\gamma = 1/\sqrt{2}$ $\quad \langle j | H | j \pm 1 \rangle = 1$

Want $\langle 2n | e^{-iHt} | 1 \rangle$

Go to infinite line



$$G(j, k, t) = \langle k | e^{-iHt} | j \rangle$$

$$= (-i)^{k-j} J_{k-j}(2t)$$

Front moves with speed 2

Go to half line

$$\tilde{G}(j, k, t) = G(j, k, t) - G(j, -k, t)$$

Go to Finite Segment

EXACT ANSWER

$$\tilde{G}(j, k, t) = \sum_{l=-\infty}^{\infty} \Big\{ G(j, k + 2l(2n+1), t) - G(j, -k + 2l(2n+1), t) \Big\}$$

Plug In $j = 1$ $\quad k = 2n$ $\quad t \simeq n$

$\tilde{G}(1, 2n, \simeq n)$ is big!!!

Quantum Walk is exponentially faster than Classical Random walk in going from ENTRANCE to EXIT.

Can we use This to solve a problem?

Suppose Graph is given in terms of an ORACLE

Nodes have random names given as strings of 2n bits

You give ORACLE a name

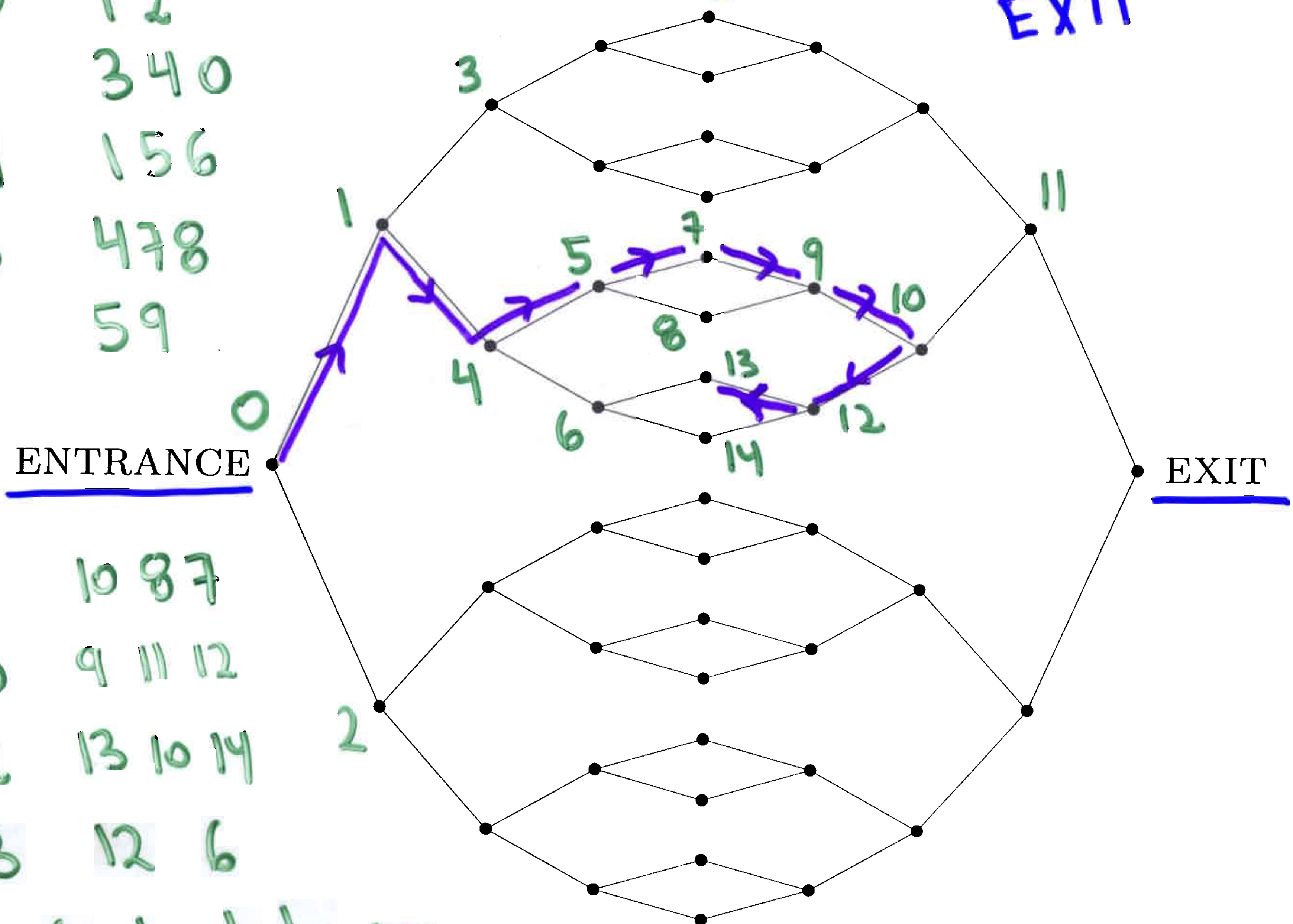
If name corresponds to a valid node then ORACLE outputs names of connected nodes

Problem: Given an ORACLE for the graph and the name of ENTRANCE find the name of EXIT

Classical Algorithm Can find EXIT

13

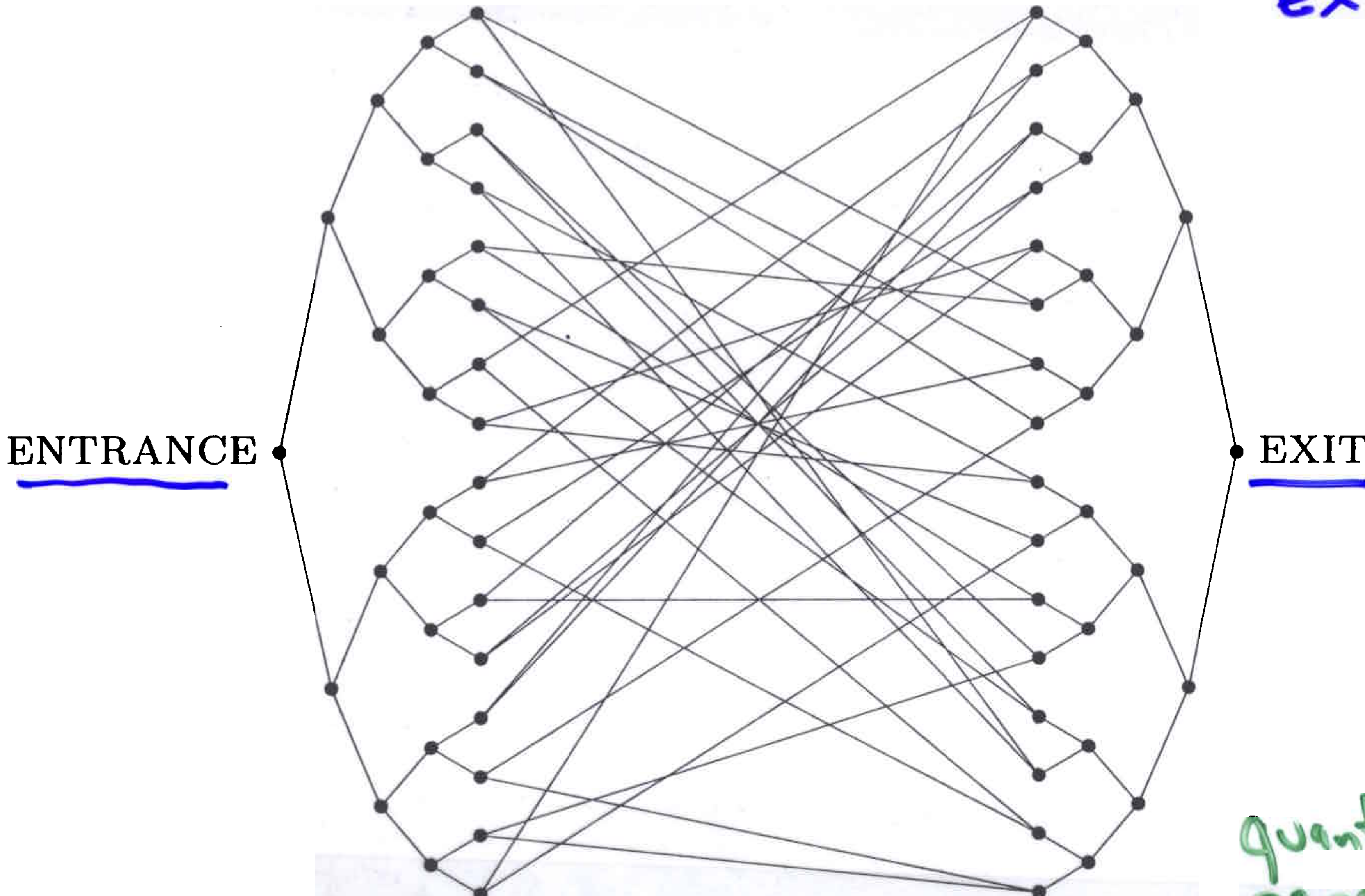
In	Out
0	1 2
1	3 4 0
4	1 5 6
5	4 7 8
7	5 9
9	10 8 7
10	9 11 12
12	13 10 14
13	12 6



Go back to 10

Join two Binary trees with a Random Cycle

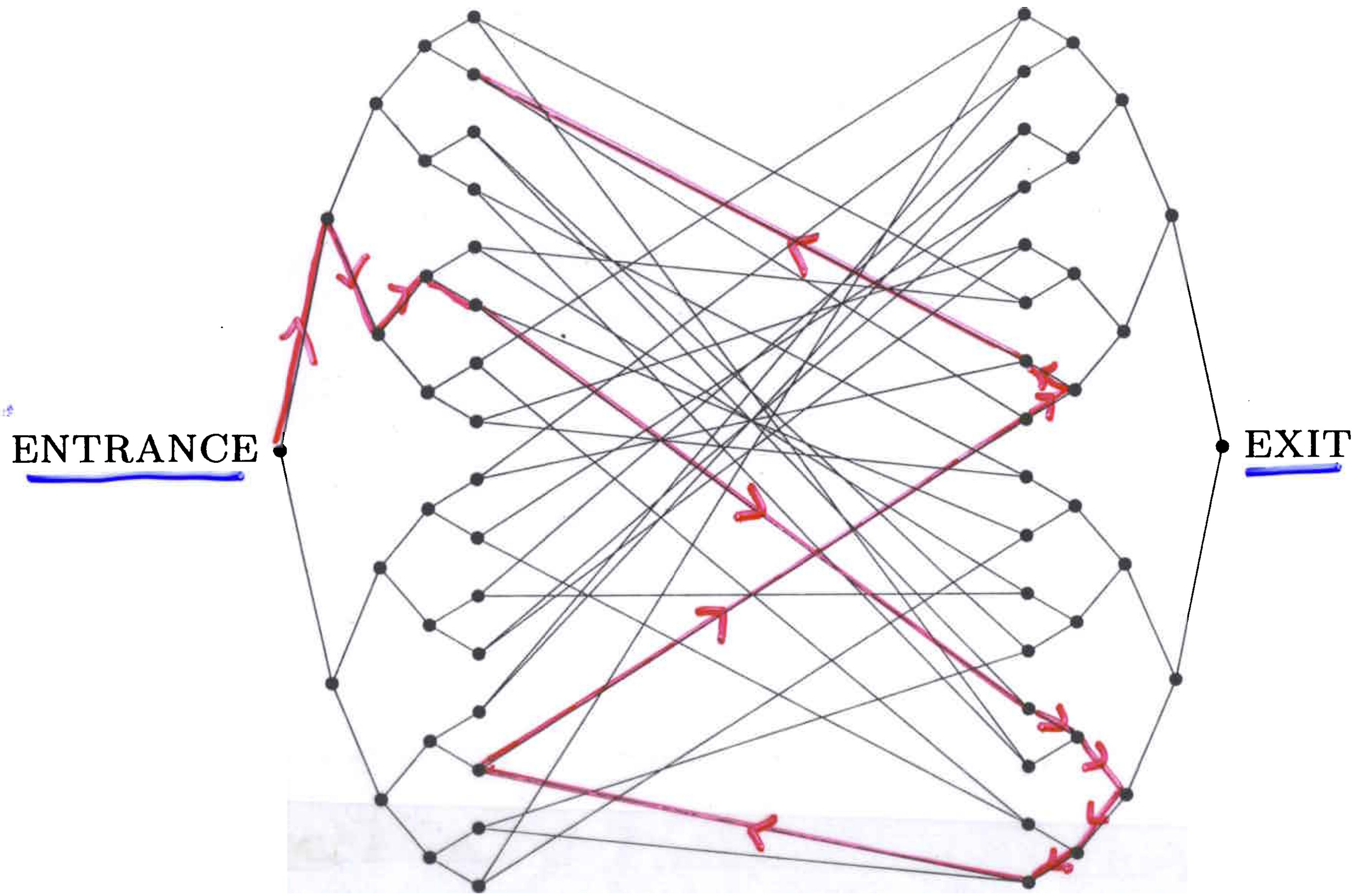
Valence is 3 at all nodes except ENTRANCE and EXIT



quant-ph/0209131

Childs, Cleve, Deotto, Farhi, Gutmann, Spielman

→ Nodes queried by algorithm

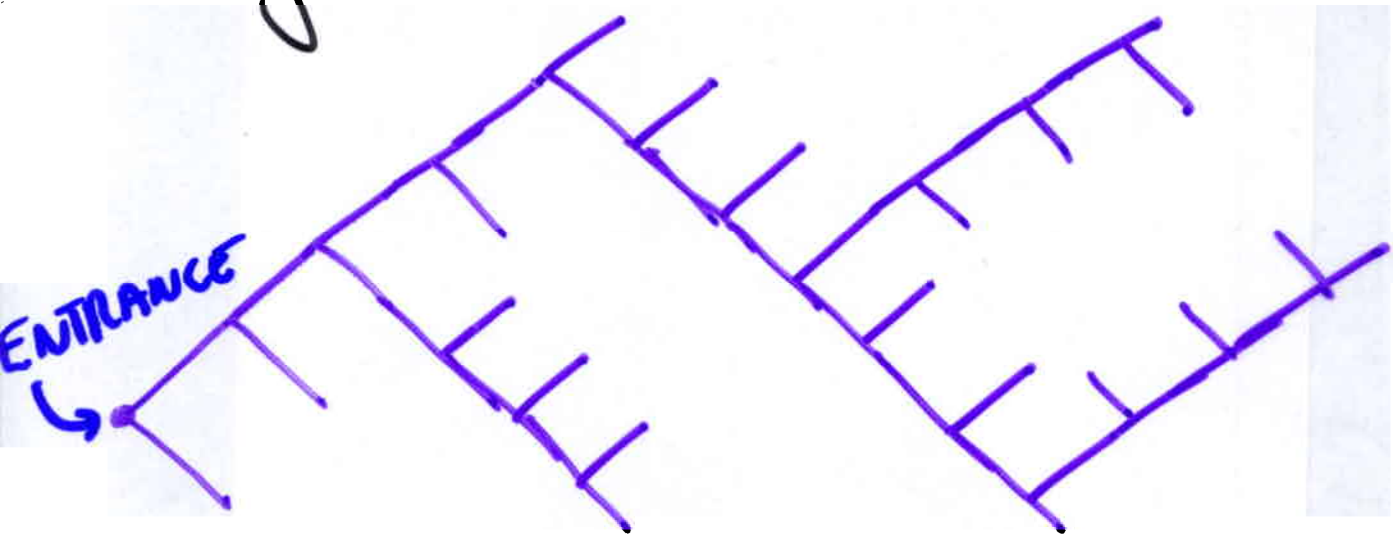


Each vertex has degree 3

You can only tell if you are in the middle if you see a node again (cycle)

This is exponentially unlikely

History of Algorithm is a tree - with no cycles - that does not hit EXIT



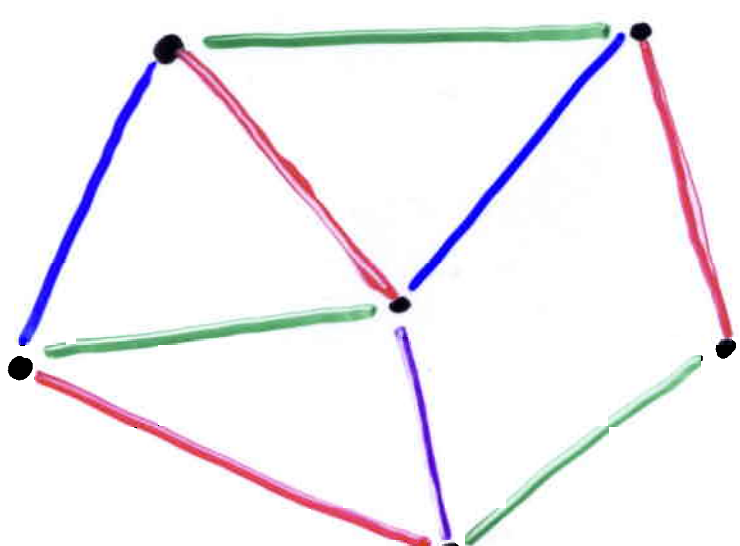
Theorem Any Classical Algorithm that makes at most $2^{n/6}$ queries to the ORACLE finds EXIT with probability at most $4 \cdot 2^{-n/6}$

Quantum Walk

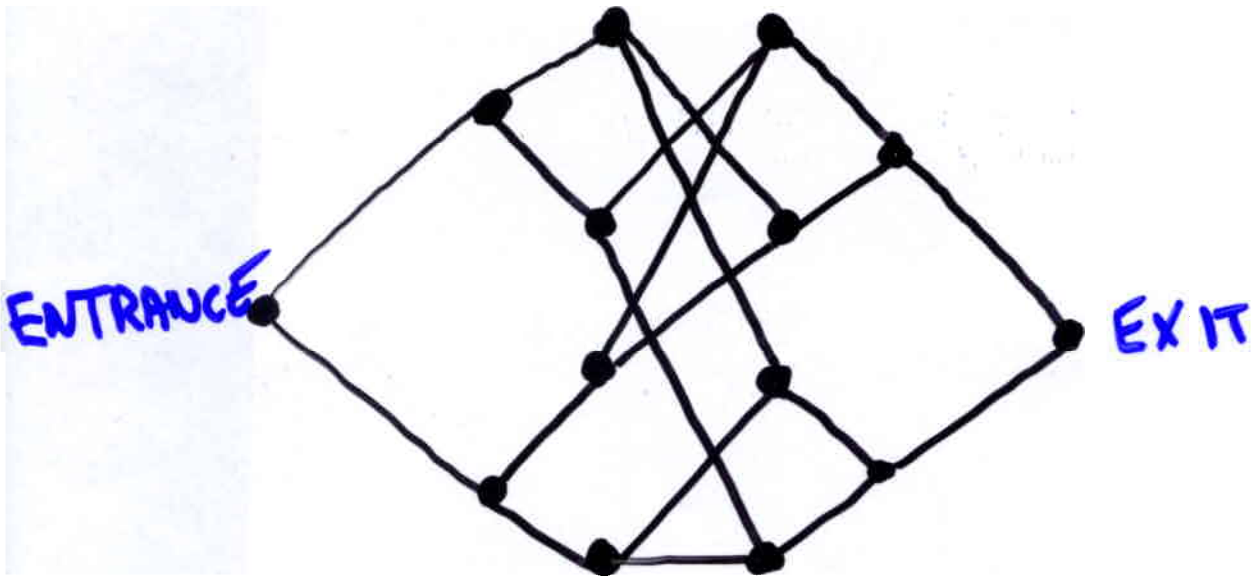
Promote classical ORACLE to a quantum operator $\hat{O}$

$$\boxed{e^{-iHt} \simeq \hat{V}_k \hat{O} \hat{V}_{k-1} \hat{O} \ldots \hat{O} \hat{V}_1}$$

We showed how to do this for a general Graph G which also has a consistent coloring of the edges — No vertex has two edges with same color



Recent Results: Colors not needed!!

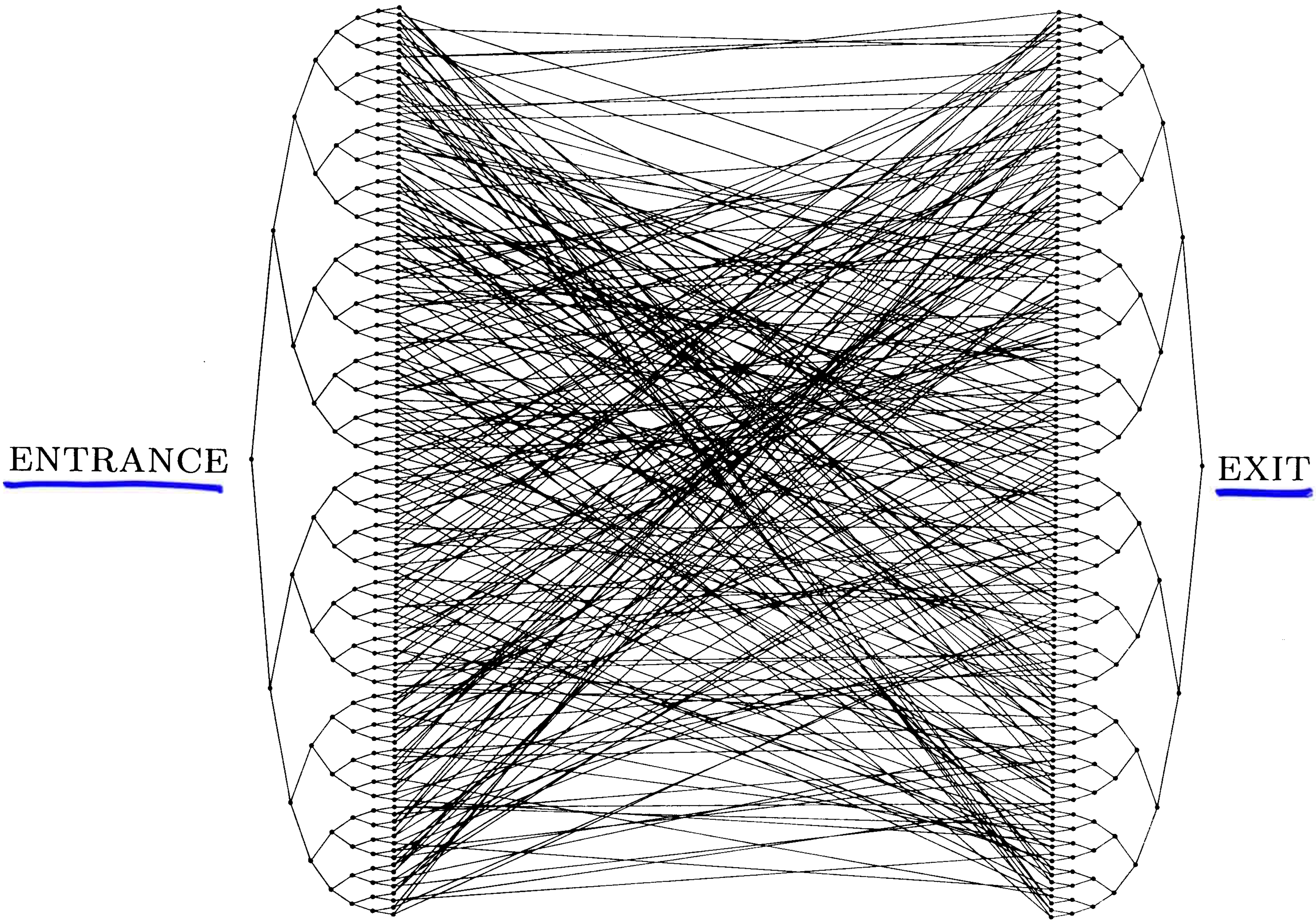


Column space is still Invariant !!

Walk is on a finite segment with a defect in the middle

$$\boxed{\text{Go from ENTRANCE to EXIT with polynomial number of queries}}$$

Quantum Algorithm does not provide a path from ENTRANCE to EXIT !!



ENTRANCE
EXIT