Eulerian Circuits

Quantum Algorithms for Graph Traversals

Sasha Levinshteyn

Outline

Housekeeping

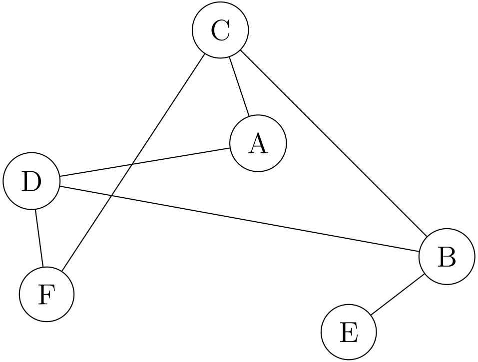
- Join the Discord!

Section 1

Graph Definitions and Problems

Definition of a Graph

- A graph $G = (V, E)$ consists of
  - A set of vertices V with $|V| = n$
  - Binary relations/edges E between vertices with $|E| = m$

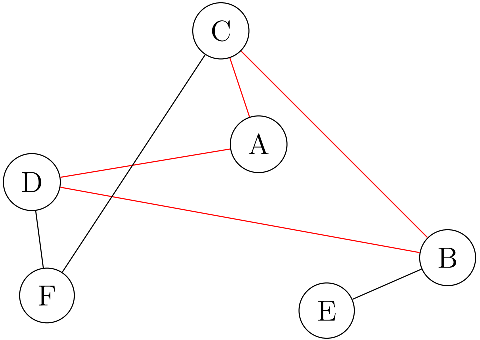


Degrees and Adjacency

- If two vertices u and v have an edge between them, they are call *adjacent*.
- The *degree* of vertex v is the number of edges with v as an endpoint or the number of vertices adjacent to v, denoted $d(v)$.
- In a directed graph, the *out-degree* of a vertex v is the number of edges coming out of v, denoted $d^+(v)$.
- Similarly, the *in-degree* of a vertex v is the number of edges coming into v, denoted $d^-(v)$.

Cycles

- A *cycle* is a sequence of vertices $(v_1, v_2, \ldots, v_k, v_1)$ such that
  - There is an edge (v_i, v_{i+1}) for $1 \leq i \leq k-1$
  - There is an edge (v_k, v_1)
- A cycle of k vertices is called a k-cycle.

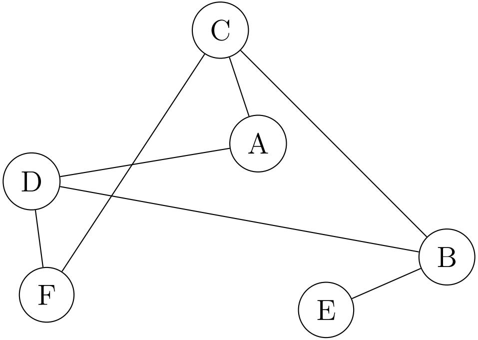


Representations of Graphs

- How do we actually represent a graph mathematically or in an algorithm?
- *Adjacency Matrix Model*
  - Given $n \times n$ matrix $A \in \{0, 1\}^{n \times n}$ with $A_{ij} = 1$ if there is an edge between vertices i and j and $A_{ij} = 0$ otherwise.
- *Adjacency List Model*
  - Given degrees $d(v)$ of each vertex v.
  - Given an array $n(v)$ for each vertex v.

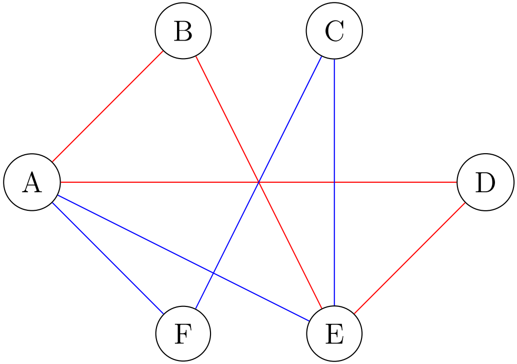
Representations of Graphs

- Let's do this one together!



Eulerian Circuits

- An *Eulerian circuit* or *Eulerian tour* is a closed walk (cycle that can repeat vertices) that uses every edge exactly once.



Eulerian Circuit Problem

- *Eulerian Circuit Problem*: Determine if a graph G has an Eulerian circuit (and is thus Eulerian).
- Unlike the Hamiltonian cycle problem from last week, this is very much NOT NP-hard.
- As it turns out, even classically, we can do this in $O(n^2)$ time in the adjacency matrix model and $O(n)$ time in the adjacency list model.

Euler's Theorem (The Graph Theory One (The Eulerian Circuits One))

- A (connected undirected) graph has an Eulerian circuit if and only if every vertex has even degree.

Euler's Theorem Proof

- Assume that we have a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$.
- Assume G has an Eulerian circuit $v_1 \to v_2 \to \cdots \to v_m \to v_1$, where the vertices may repeat but the edges may not.
  - The total amount of vertices (repeats included) in the circuit must be equal to the total amount of edges in the graph.
  - The degree of a vertex is equal to the number of edges connected to it.
  - We can go through the circuit and increment the degree of a vertex each time we encounter an edge that uses that vertex.
  - Each time we encounter a vertex, we encounter exactly 2 adjacent edges, which must all be unique since this is an Eulerian circuit.
  - So, each vertex in G has even degree.

Euler's Theorem Proof

- Assume that we have a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$.
- Assume all vertices of G have even degree.
  - Induct on the number of vertices n.
  - Base Case: A graph with 1 vertex and thus 0 edges vacuously has an Eulerian circuit.
  - Assume that every graph with $k < n$ vertices all with even degree has an Eulerian circuit.

Euler's Theorem Proof

- Assume all vertices of G have even degree.
  - Now, we have n vertices.
  - Choose some vertex v_0 and remove it and its adjacent edges to create a new graph G'.
  - We must have removed an even amount $2x$ of edges to an even amount of other vertices $v_1, v_2, \ldots, v_{2x-1}, v_{2x}$.
  - Split these other vertices into x pairs (v_{2i-1}, v_{2i}).
  - For each pair, if they were already connected with an edge, remove that edge, otherwise, add that edge.
  - Now, each of the affected vertices in G' has either lost one edge and gained one edge or lost two edges, maintaining the even degree property.

Euler's Theorem Proof

- Assume all vertices of G have even degree.
  - G' is now composed of some number of connected components, each containing only vertices of even degree.
  - By the inductive hypothesis, each of these connected components has an Eulerian circuit.
  - Consider the pairs of vertices (v_{2i-1}, v_{2i}) we connected.
  - If we remove the edge and add v_0 and its adjacent edges again, we can go from v_{2i-1} to v_0 to v_{2i} instead.
  - Consider the pairs of vertices (v_{2j-1}, v_{2j}) we disconnected.
  - If we readd the edge and add v_0 and its adjacent edges again, we can go from (the Eulerian circuit of the CC v_{2j-1} was a part of) to v_{2j-1} to v_0 to v_{2j} to (the Eulerian circuit of the CC v_{2j} was a part of if different) to v_{2j-1}, reconstructing an Eulerian circuit in G.

Questions?

Section 2

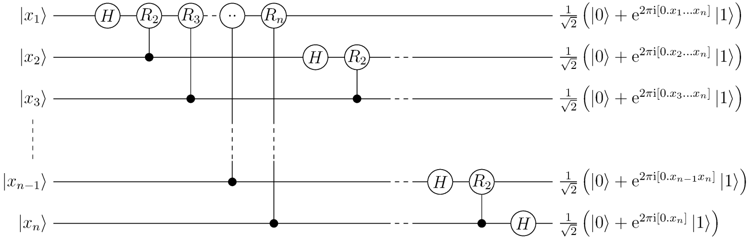
Quantum Computing Speedrun

Motivation

- I firmly believe we should add the word quantum in front of everything, obviously!
- If $P = NP$, classical cryptography breaks but quantum cryptography...
  - Come to SIGQuantum tomorrow at 6 pm to find out!
- Entanglement and superposition can be exploited to store and manipulate data, exploring multiple paths simultaneously, in ways that classical computers can't
- We can (theoretically) get much faster runtimes for algorithms that require certain kinds of computation

Quantum Algorithms

- A *quantum algorithm* is represented by a *quantum circuit* that...
  - Acts on some amount of input *qubits*
  - Composed of *quantum gates* that transform the qubits
  - Ends with a *measurement* on some of the qubits



[Tre18]

Qubits, Gates, and Other Scary Words

- A qubit is a **vector**!
  - When working with a single qubit, the standard basis vectors are

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

  - A given unit vector can be represented as a linear combination:

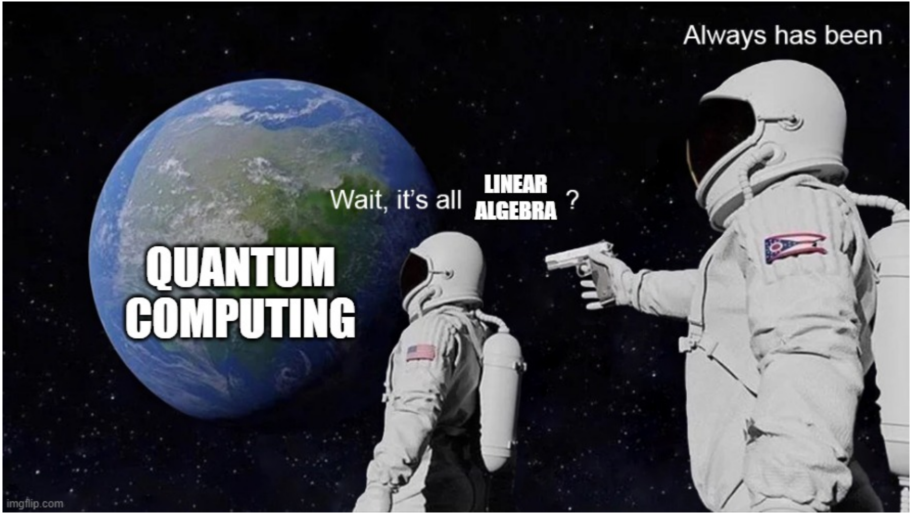
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, |\alpha|^2 + |\beta|^2 = 1, \alpha, \beta \in \mathbb{C}$$

  - When measured in the standard basis, the qubit will collapse into the $|0\rangle$ state with probability α^2 or the $|1\rangle$ state with probability β^2

Qubits, Gates, and Other Scary Words

- A gate is a **matrix**!
- Most gates that you will encounter are...
  - Unitary matrices: $UU^\dagger = U^\dagger U = I$
  - Hermitian matrices: $H = H^\dagger$
- Quantum algorithms are largely just matrices being multiplied to input vectors in ways that compute many values simultaneously.
- Don't let the physicists scare you!

Qubits, Gates, and Other Scary Words



Questions?

Section 3

Quantum Algorithms

Note About Quantum Algorithms

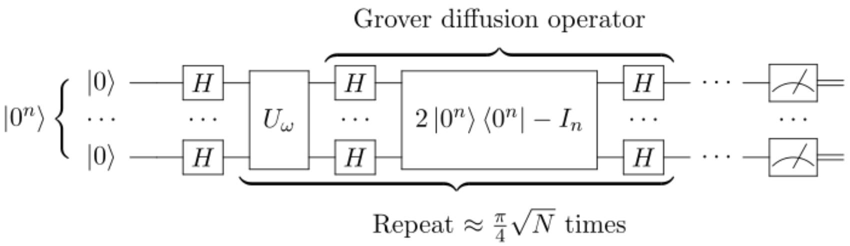
- You don't have to know many details of quantum computing to understand most quantum algorithms.
- Most quantum algorithms for graph theory problems do most of the work by calling the same few critical quantum algorithms.
- If you know what those smaller quantum algorithms do, you can blackbox them away.

Notions of Complexity

- The *quantum query complexity* of a graph algorithm is the number of queries it makes to the adjacency matrix or list representing the graph.
- The *quantum time complexity* of a graph algorithm is the number basic quantum operations it makes.

Grover's Algorithm

- Quantum Search Problem: Suppose we have a function $f : \{0, 1, \ldots, n-1\} \to \{0, 1\}$ such that $f(x) = 1$ for at most one x, which we will call ω. Find this x if it exists.
- Grover's algorithm can do this using $O(\sqrt{n})$ evaluations of f, beating the classical $O(n)$ evaluations.
- $U_\omega |x\rangle = (-1)^{f(x)} |x\rangle$



Reducing Error

- Quantum algorithms are not always correct.
- They output an incorrect answer with a probability p.
- To get the probability of an incorrect answer down to ϵ, we need to repeat r times so that $p^r \leq \epsilon$, so we can make the probability of an incorrect answer less than $\frac{1}{n}$ by repeating each quantum subroutine $r = O(\log n)$ times
- So, if we have a quantum algorithm with $O(f(n))$ quantum query complexity, it will typically have $O(f(n) \log n)$ quantum time complexity.
- Grover's algorithm actually needs an extra logarithmic factor, so it will have $O(\sqrt{n} \log^2 n)$ quantum time complexity.

Section 4

Eulerian Circuit Problem Quantum Algorithm

Adjacency List Model

- Algorithm [rn07]
  - The degree of every vertex is given.
  - Search the list of degrees for an odd number using a simple quantum search.
  - If we find an odd number, it is not Eulerian.
  - Otherwise, it is Eulerian.
- Complexity
  - The quantum search requires $O(\sqrt{n})$ queries to the list of degrees, making this algorithm have $O(\sqrt{n})$ quantum query complexity.
  - As this search utilizes Grover's algorithm, the algorithm will have $O(\sqrt{n}\log^2 n)$ quantum time complexity.

Adjacency Matrix Model

- Algorithm
  - This time, we don't know the degrees right away.
  - Grover's algorithm in combination with a classical algorithm can be used to compute the parity of each column.
  - If we find an odd number, it is not Eulerian.
  - Otherwise, it is Eulerian.
- Complexity
  - This algorithm will have $O(n^{1.5})$ quantum query complexity.
  - As this search utilizes Grover's algorithm, the algorithm will have $O(n^{1.5} \log^2 n)$ quantum time complexity.

Brainteaser

13 Apples, 15 Bananas and 17 Cherries are put in a hat. Inside the hat, if two fruits of different type collide, they both get converted into the third type. For example 1 apple and 1 banana can collide to form 2 cherries. Can a sequence of collisions lead to all 45 fruits having just one type?

Bibliography I

Fawly.

Grover's algorithm circuit.

https://commons.wikimedia.org/wiki/File:Grover%27s_algorithm_circuit.svg, 2021.

Accessed: 09-15-2024.

Sebastian Dö rn.

Quantum algorithms for optimal graph traversal problems.

https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Mitarbeiter/doern/GT.pdf, 2007.

Accessed: 09-15-2024.

Trenar3.

Quantum circuit for quantum-fourier-transform with n qubits.

https://commons.wikimedia.org/wiki/File:Q_fourier_nqubits.png, 2018.

Accessed: 09-15-2024.