Discrete Structures II

Mwangi H. (Ph.D.)

CS Yr 3.1
Department of Computing
J.K.U.A.T.

April 9, 2024

Outline

Graph Theory

- Graphs are not plots - function plots nor is it "a graph of function".
- A graph is one of the most foundational and useful data structures in computer science.
- If there's two things in discrete mathematics to take into the rest of your computer science studies it is induction and graphs.

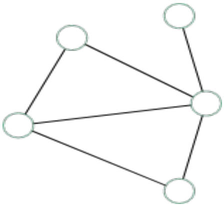


Figure 1.1: A graph with 5 vertices and 6 edges

Graph Theory

- In its simplest form, a graph comprised of vertices(nodes, dots, circles, squares, shapes), connected by edges(lines between vertices).
- Graphs are incredibly useful in practice and have countless applications.
  1. Computers in a network (e.g. the internet)
  2. Destinations in a transport network (e.g. bus stops, airports)
  3. Social connections (e.g. friends on Facebook)
  4. Semantic and conceptual connections (e.g. fish is “related” to ocean)
  5. Components of a software system (e.g. classes, modules, libraries)

Graph Theory

- Some formal definitions

Definition 1.1: Graph

A graph is a pair (V, E) consisting of non-empty set of vertices V and a set of edges E. E is a subset of the set $\{\{u, v\} \mid u, v \in V\}$

- V can be any set of discrete elements. Often, it is a subset of the natural numbers. On the other hand, the definition of E needs some parsing.E is a set of subsets of V. i.e. $E \subset \mathcal{P}(V)$
- Each of the element of E has
  1. two elements. e.g. $\{u, v\}$ where $u, v \in V$ or
  2. one element, e.g. $\{v, v\} = \{v\}$ for some vertex $v \in V$

Graph Theory

- A graph of 5 vertices

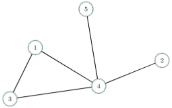


Figure 1.2: A graph with 5 vertices

- Fig. 1.2 represent a graph with five vertices and 5 edges

$$V = \{1, 2, 3, 4, 5\} \qquad E = \{\{1, 3\}, \{3, 4\}, \{1, 4\}, \{4, 5\}, \{2, 4\}\}$$

Example 1.1

Draw the graph (V, E) for $V = \{1, 2, 3, 4\}$ and $E = \{\{1, 2\}, \{3, 4\}, \{4, 1\}\}$

The Language of Graphs

- Important terminologies with regards to graphs
  1. A **simple graph** is what we have defined already as a graph. There is at most one edge between any two vertices.
  2. A **multigraph** is what we call a graph which allows multiple edges between vertices.
  3. A **node** or **vertex** is a discrete object of the graph.
  4. The two vertices of an edge are called **endpoints**. That edge is said to **join** or **connect** the two vertices and the edge is **incident** to each of the vertices it joins.
  5. When two vertices are joined by an edge, those vertices are called **adjacent**.
  6. An edge which connects a vertex to itself is called a **loop**.

The Language of Graphs

- More formal definitions about graphs

Definition 1.2: Order of a Graph

A graph $G = (V, E)$ of order n has $|V| = n$

Definition 1.3: Neighbourhood

The neighbourhood of a vertex v, $N(v)$, of a graph $G = (V, E)$ is the set of all vertices adjacent to v in G

$$N(v) = \{u \mid \{u, v\} \in E\}$$

The Language of Graphs

Definition 1.4: Degree of a Vertex

The degree of a vertex v, $deg(g)$, is the number of edges incident with it. Note that a loop contributes 2 to its degree.

Theorem 1.1: The Handshake Theorem

If $G = (V, E)$ is a graph of m edges, then

$$2m = \sum_{v \in V} deg(v)$$

The Language of Graphs

Proof of Theorem 1.1.

Each edge contributes twice to the degree count of all the vertices in the graph. Indeed, each edge is incident to exactly two vertices.
$\{u, v\} \in E$ contributes 1 to both $deg(u)$ and $deg(v)$
Thus, it must be that $\sum_{v \in V} deg(v)$ is twice the total number of edges. □

- The handshake theorem actually leads to a very interesting theory about graphs.

Theorem 1.2

A graph $G = (V, E)$ always has a an even number of vertices with odd degree.

The Language of Graphs

Proof of Theorem 1.2.

Partition the set of vertices V into V_1, all vertices with even degree, and V_2 all vertices with odd degree. By the handshake theorem we have:

$$\begin{aligned} 2m &= \sum_{v \in V} deg(v) \\ &= \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v) \end{aligned}$$

The left-hand side of this equation, $2m$ is obviously even. Since each v in V_1 has even degree, and the sum of even numbers is even, then $\sum_{v \in V_1} deg(v)$ is also an even number.

The only way for an even number plus another number to be equal to an even number is if the other number is even. Thus, $\sum_{v \in V_2} deg(v)$ is an even number.

But, each v in V_2 has odd degree. How can the sum of odd numbers be even? Only when the number of numbers in the sum is even. Thus, there must be an even number of vertices with odd degree. $\square$

- Just us there are subsets, we also have subgraphs

Definition 1.5: Subgraph

A subgraph of a graph $G = (V, E)$ is another graph (W, F) such that $W \subseteq V$ and $F \subseteq E$. A proper subgraph of G if $W \subsetneq V$

- In this case we require the sets W and F to form a graph. Therefore, there cannot be edges in F whose endpoints are not also in W
- In fig.1.3, the leftmost graph G_1 is a simple graph with 5 vertices and 6 edges. $\{1, 2, 3, 4, 5\}$ and $\{\{1, 2\}, \{2, 4\}, \{4, 5\}, \{5, 3\}, \{3, 2\}, \{2, 5\}\}$
- The G_2 in fig. 1.3 is a subgraph of G_1. However G_3 is not a graph - one edge has no endpoint.

Sub-Graphs and Induced Graphs

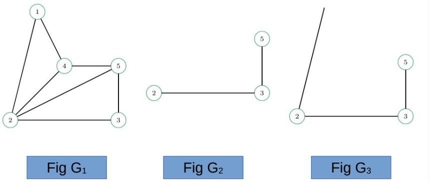


Figure 1.3: Subgraph and Induced Graph

- A special kind of subgraph is an induced subgraph

Sub-Graphs and Induced Graphs

Definition 1.6: Induced Graph

For a graph $G = (V, E)$ and a set of vertices $W \subseteq V$, a subgraph induced by W is the graph (W, F) where an edge is in F if and only if $\{u, v\} \in E$ for $u, v \in W$

In fig. 1.3 - G_2 is not an induced subgraph because it is missing the edge $\{2, 5\}$ of the original graph G_1

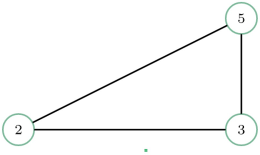


Figure 1.4: The subgraph induced by $\{2, 3, 5\}$ of G_1 of fig. 1.3

Connectivity

- Arguably the most important thing about graphs is that they encode connections.
- Recall that the definition of a graph only requires vertices and connections (edges) between them.
- We don't care about the placement of vertices or the length of edges, or even if edges overlap.
- All we care about in a simple graph is whether two vertices are adjacent or not.
- Understanding whether two vertices are connected is the study of connectivity.

Definition 1.7: Path

A path of a simple graph $G = (V, E)$ is a sequence of vertices (v_n) where an edge exists between v_i and v_{i+1} for $1 \leq i < n$

Connectivity

Definition 1.8: Connectivity

Two vertices u, v in a graph are connected if there exists a path from u to v. Otherwise u and v are said to be disconnected. A graph is connected if every pair of vertices in the graph is connected.

Other graph terminologies.

1. When a path begins and ends at the same vertex it is called a **circuit**
2. A path $(v_1, v_2, \ldots, v_n)$ is said to **pass through** the vertices $v_2, v_3, \ldots, v_{n-1}$ and **traverse** the edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$
3. A path is **simple** if every edge in the path only appears once.
4. When a path exists between u and v, we say that v is **reachable** from u

Graph Isomorphisms

- Whether two graphs are actually the exact same, but just drawn differently, or are "almost" the same, we can formally describe graphs as being "sufficiently similar" by isomorphisms.

Definition 1.9: Isomorphic Graphs

Two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection function f from V_1 to V_2 with the property that two vertices $v_1, u_1 \in V_1$ are adjacent in G_1 if and only if $f(v_1)$ and $f(u_1)$ are adjacent in G_2. The bijective function f is called an isomorphism.

- From the definition of isomorphic graphs, it is obvious that two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ must have $|V_1| = |V_2|$ and $|E_1| = |E_2|$ to be isomorphic. But this is not enough, adjacencies must be maintained as well.

Graph Isomorphisms

- The following two fig. 1.5 G_1 and G_2, we can find a bijection between these graphs

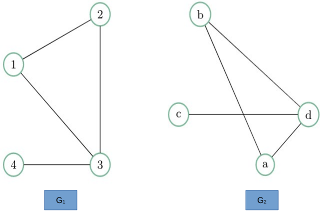


Figure 1.5: Two isomorphic graphs

- $$f(1) = a, \quad f(2) = b, \quad f(3) = d, \quad f(4) = c$$

  It works because

  1. 1 is connected to 2 and 3 in G_1 and $f(1) = a$ is connected to $f(2) = b$ and $f(3) = d$ in G_2

Graph Isomorphisms

2. 2 is connected to 1 and 3 in G_1 and b is connected to a and d in G_2
3. 3 is connected to $1, 2,$ and 3 in G_1 and d is connected to a, b and $f(4) = c$ in G_2
4. 4 is connected to 3 in G_1 and c is connected to d in G_2

Complete Graphs

- A complete graph is a special kind of connected graph.
- Not only must the graph be connected—there must be a path from every vertex toe very other vertex—but each path must be of length 1.
- That is, every vertex must be adjacent to every other vertex.

Definition 1.10: Complete Graph

A complete graph is a graph where every one of its vertices is adjacent to every other vertex. A complete graph of order n is denoted by K_n

Complete Graphs

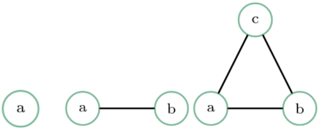


Figure 1.6: The complete graphs K_1, K_3, K_3

- A graph containing a single vertex is complete (vacuously so).
- A graph containing two vertices connected by a single edge is also complete.
- A graph of three connected into a triangle is also complete.

Complete Graphs

Theorem 1.3

A complete graph of order n has $\frac{n(n-1)}{2}$ edges.

Proof.

A complete graph has an edge between any two vertices. For a graph with n vertices, if we choose any two vertices there must be an edge between them. There are $\binom{n}{2}$ such choices

$$\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{(n-1)(n)}{2}.$$

□

Cliques

- Complete graphs are very special graphs. There's not many of them. Indeed, for a positive integer n there is exactly one complete graph of order n upto isomorphism. It is K_n

Definition 1.11: Clique

For a graph $G = (V, E)$ a clique is an induced subgraph of G that is complete.

Definition 1.12: Maximal Clique

A maximal clique $G = (V, E)$ is a clique that cannot be made larger by including any other adjacent vertex of V

Cliques

Example 1.2: Cliques

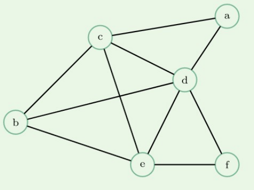


Figure 1.7: Find all cliques of size 3 or more in this figure. Which are maximal?

1.$\{a, c, d\}$ 2.$\{d, e, f\}$ 3.$\{b, c, d, e\}$

4.$\{b, c, d\}$ 5.$\{b, d, e\}$ 6.$\{c, d, e\}$ 7.$\{b, c, e\}$

$\{a, c, d\}$, $\{d, e, f\}$, $\{b, c, d, e\}$ are maximal

Cycles

Definition 1.13: Cycle Graph

A cycle graph is a graph with exactly one cycle. The cycle graph of order n is denoted C_n

As a consequence of cycles, a cycle graph must have 3 or more vertices.

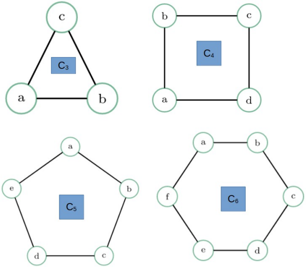


Figure 1.8: Cycles and their orders

Wheels

Definition 1.14: Wheels

A wheel graph is a cycle graph in which one extra vertex has been added which connects to every other vertex in the cycle. The wheel graph of order n is denoted W_n

Definition 1.15: Planar Graphs

A planar graph is a graph that can be drawn with no overlapping edges.

Planar graphs have many nice properties and theorems associated with them. Most of the graphs we've seen so far are planar. Every cycle graph is planar. Every wheel graph is planar. In contrast, every complete graph of order 5 or higher is not planar.

Trees

- Trees are an extra special type of graph

Definition 1.16: Tree

A tree is a simple connected graph with no cycles.

Definition 1.17: Rooted Tree

A rooted tree is a tree in which one of the vertices has been designated as the root.

- Why are rooted trees important? Because it gives us a canonical view of the tree.

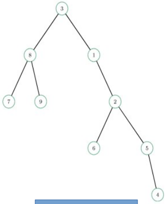
- Designating a root in a tree induces special relationships between vertices.
- It derives from that fact that there is a unique simple path from any vertex to any other vertex in a tree. The terminology used most commonly is based on ancestry.
  - The root is the “oldest” ancestor.
  - Each of the **children** of the root are the vertices which have a path of length 1 from the root.
  - The children of those children are vertices which have a path of length 2 from the root.
  - Conversely, the children of the root tree have the root as their **parent**.
  - To speak about a vertex’s children and children’s children, etc. we say **descendants**.

Binary Trees

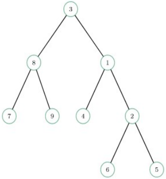
- A binary tree is perhaps the most important tree structure in computer science.

Definition 1.18: Binary Tree

A binary tree is a rooted tree where each vertex has at most 2 children. A full binary tree is a binary tree where every vertex has exactly 2 children or 0 children



A binary tree.



A full binary tree.

Binary Trees

- Thinking back to Recursive Definitions, binary trees lend themselves very naturally to a recursive definition.
- Consider the following recursive definition of the set of all possible full binary trees.

Example 1.3: Binary Tree Recursive

1. Basis Step: A simple vertex r is the root of a binary tree
2. Recursive Step: Let r be a new vertex which is not in T_1 or T_2. Add an edge from r to r_1 and from r to r_2. The resulting graph is a full binary tree rooted at r

Structural Induction

- We have already seen mathematical induction, strong induction, and the well-ordering principle.
- Now, structure induction is a specialization of mathematical induction to the context of recursively-defined structures.
- We've seen recursively defined sets previously, and we have just seen recursively defined binary trees.
- The goal now is to use induction to prove some properties on these recursive definitions.
- Much like mathematical induction, structural induction has a basis step, an inductive step, and an inductive hypothesis.

Structural Induction

Definition 1.19: Struction Induction

1. Basis step: Show that the desired property is true for all elements specified in the basis step of the recursive definition.
2. Inductive step: Assume that the desired property is true for all of the old elements which are used to construct new elements. Then, show that the desired property must also hold for the new elements.

1. For the fig. 1.10 find all simple paths from 2 to 4 which do not traverse 2 or 4

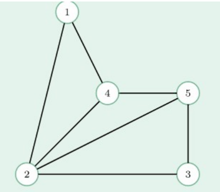


Figure 1.10: Exercise on Graphs

Directed Graphs

- In a graph, we only care about whether vertices are connected or not; an edge is a set $\{u, v\}$ indicated that the vertices u and v are adjacent.
- If we want the connection to be ordered, then we arrive at directed graphs or digraphs. The only difference from simple graphs is that edges are now ordered pairs (u, v)

Definition 2.1: Directed Graph

A directed graph is a pair (V, E) consisting of a non-empty set of vertices V and a set or directed edges E with $E \subseteq V \times V$

- A directed edge is sometimes called an **arc**.
- A directed edge (u, v) starts at u and ends at v. In other words, the directed edge has **initial vertex** u and **terminal vertex** v

Directed Graphs

- Our previous graph definition is also called a undirected graph.
- Note that “simple directed graphs”, “directed multigraphs”, “simple (undirected) graphs”, and “(undirected) multigraphs” are all different.

Definition 2.2: In-Degree

The in-degree of a vertex v in a directed graph is the number of edges which terminate at v It is denoted $deg^{-}(v)$

Definition 2.3: Out-Degree

The out-degree of a vertex v in a directed graph is the number of edges which start at v. It is denoted $deg^{+}(v)$

Directed Connectivity

- From undirected graphs, we say two vertices u and v are connected if the edge $\{u, v\}$ exists in the graph. And a graph is connected when a path exists from every vertex to every other vertex.
- For directed graphs, we have two notions of connectivity, since a vertex connected to an edge may be the initial vertex or the terminal vertex. In particular, the notion of a path changes.

Definition 2.4: Directed Path

A directed path (or simply path) of a directed graph $G = (V, E)$ is a sequence of vertices (v_n) where $(v_i, v_{i+1}) \in E$ for $1 \leq i < n$

- We thus have the first kind of connectivity for a directed graph, which is the same definition (although different meaning) for a directed graph.

Directed Connectivity

Definition 2.5: Strongly Connected

A directed graph is strongly connected if there is a directed path from every vertex to every other vertex.

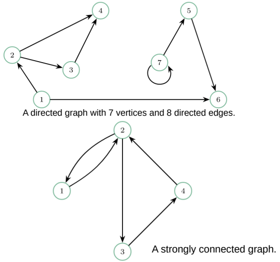


Figure 2.1: directed graph and strongly directed graphs

Binary Relations as Graphs

- Recall, Let $\mathcal{R}$ be a binary relation on a set A. We know that $\mathcal{R}$ is a subset of $A \times A$
- We also saw in the previous section that directed graphs have the set of edges E be a subset of $V \times V$, for the vertex set V. Thus, the set of edges in a directed graph actually induces a binary relation on the set of vertices.
- If you can describe the edges of a directed graph as pairs of vertices, then you can describe the binary relation induced on the vertex set.
- Consider the following graph on the vertex set $\{1, 2, 3, 4, 5, 6\}$. What is the binary relation that it encodes in the fig: 2.2?

Binary Relations as Graphs

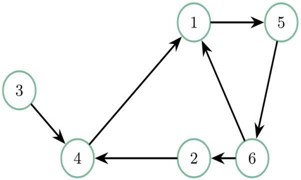


Figure 2.2: A directed graph on $\{1, 2, \ldots, 6\}$

Solution

Binary Relations as Graphs

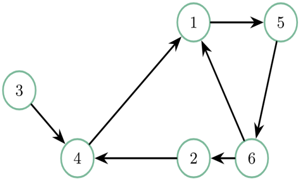


Figure 2.2: A directed graph on $\{1, 2, \ldots, 6\}$

Solution

$\mathcal{R} = \{(1, 5), (2, 4), (3, 4), (4, 1), (5, 6), (6, 1), (6, 2)\}$

Binary Relations as Graphs

Example 2.1: A graph of Binary Relation

Consider the binary relation $\mathcal{R}$ on A $A = \{1, 2, 3, \ldots, 8\}$ defined as

$\mathcal{R} = \{(1, 2), (1, 3), (2, 4), (3, 4), (4, 1), (4, 6), (5, 6), (6, 2), (6, 7), (7, 8), (8, 5)\}$

Can you draw the corresponding graph?

Solution

Binary Relations as Graphs

Example 2.2: A graph of Binary Relation

Consider the binary relation $\mathcal{R}$ on A $A = \{1, 2, 3, \dots, 8\}$ defined as

$$\mathcal{R} = \{(1,2), (1,3), (2,4), (3,4), (4,1), (4,6), (5,6), (6,2), (6,7), (7,8), (8,5)\}$$

Can you draw the corresponding graph?

Solution

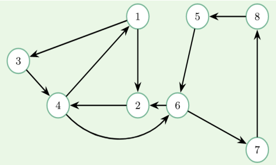


Figure 2.3: Finding connected Components

Directed Acyclic Graphs

- A very important class of directed graphs is directed acyclic graphs. As the name suggests, they are directed graphs with no loops!

Definition 2.6: Directed Acyclic Graph

A directed acyclic graph (DAG) is a directed graph with no directed cycles.

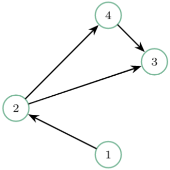


Figure 2.4: A very simple DAG

Directed Acyclic Graphs

- Indeed, DAGs have certain properties as a simple consequence of having no cycles:
  1. DAGs have no strongly connected components (other than single vertices). Only through cycles can a directed path exist from a vertex u to a vertex v and from v to u.
  2. DAGs have at least one source vertex.
  3. DAGs have at least one sink vertex.
- DAGs are incredibly important for describing scheduling problems, program structure, data processing, machine learning, version control, and much more.
- The most useful property of DAGs, and why they are so useful in practice, is that they can always be topologically ordered.
- That is, they can be drawn in such a way that all edges point in the same direction. Consider the below DAG. It is a mess of edges.

Directed Acyclic Graphs

Example 2.3: Complicated DAG

Consider the below DAG. It is a mess of edges.

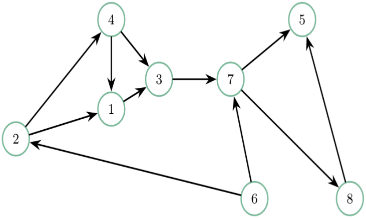


Figure 2.5: A more complicated DAG with 8 vertices.

- A topological ordering of this DAG has all edges oriented in the same direction.
- Not necessarily parallel lines or anything, but more or less “flowing” in the same direction.
- Typically, left to right, or top to bottom.

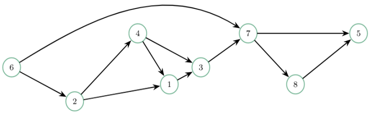


Figure 2.6: A topological ordering

Control-Flow Graphs

- Consider what the topological ordering of the DAG 2.6. It gives an explicit "prerequisite structure":
    1. 6 comes before 2 and 7
    2. 2 comes before 4 and 1
    3. 4 comes before 1 and 3
    4. 1 comes before 3
    5. 3 comes before 7
    6. 7 comes before 8 and 5
    7. 8 comes before 5
- This has natural applications to scheduling and data processing.
- DAGs are also very good at representing the control flow of programs. In particular, they can be used to represent control-flow graphs. Control-flow graphs are essential to compiler theory and program optimization.
- In general, control-flow graphs can have cycles, and are thus described best by directed graphs. However, loop management is particularly challenging.

Exercises

- Find the strongly connected components of the below directed graph.

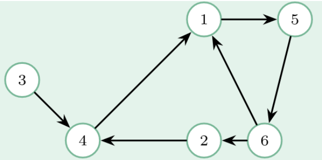


Figure 2.7: Find the strongly connected components

Representing Graphs

- how can we make use of graphs in computers? Just like integers, which have Integer Representations, graphs also have special representations on computers.

Adjacency Lists

- Adjacency lists are very simple data structures for encoding a graph. It is essentially a dictionary of lists.
- Each vertex in the graph is a key in the dictionary. The value corresponding to the key is the list of vertices adjacent to the key.

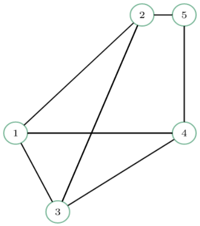


Figure 3.1: Adjacency Lists

Adjacency Lists

- The adjacency list of the graph 3.1 is:

$$\begin{aligned}&\{\\&\quad(1,\{2,3,4\}),\\&\quad(2,\{1,3,5\}),\\&\quad(3,\{1,2,4\}),\\&\quad(4,\{1,5\}),\\&\quad(5,\{2,4\})\\&\}\end{aligned}$$

- In practice, an adjacency list may be implemented as a dictionary of lists, a dictionary of sets, a list of lists, a list of sets, etc.
- It depends on the facilities of the particular programming language you are using.

Adjacency Lists

Give the adjacency list encoding the above graph. Use Python syntax if you'd like.

Example 3.1: Adjacency Lists

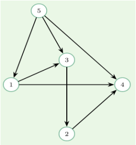


Figure 3.2: Adjacency Lists

Adjacency Matrix

- A more explicit representation compared to the adjacency list is an adjacency matrix. In this representation, every possible pair of vertices is represented in a matrix.
- The value of a particular entry in the matrix designates whether an edge exists between the pair of vertices or not.
- Give a graph $G = (V, E)$ of order n list the vertices in some order $v_1, v_2, \ldots, v_n$.
- The adjacency matrix A_G of G, with respect to this order of vertices, is an $n \times n$ zero-one matrix

$$A_G = (a_{i,j}) \qquad a_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}$$

- Adjacency matrices are best used for graphs with many edges. Otherwise, the matrix will be mostly 0s!

Adjacency Matrix

Example 3.2: Adjacency Matrix

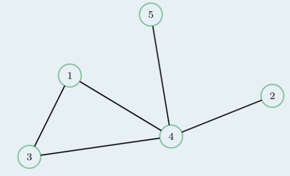


Figure 3.3: The adjacency matrix for this graph with the natural ordering $1, 2, 3, 4, 5$

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

Paths and Circuits

- We have already seen the general idea of paths, both directed and undirected.
- The study of paths in graphs is a natural extension from the basic property of adjacency between two particular vertices.
- Rather than a single edge connecting two vertices, is there a path one can traverse between the two vertices?
- You could conceive of many more practical applications of paths:
  1. Determining whether a computer on the internet is reachable;
  2. Planning a journey given specific bus/plane routes;
  3. Optimal delivery routes for couriers; or even
  4. Ecological interactions of organisms.
- There are many special kinds of paths that have received special attention in theory and practice.

Eulerian Paths and Circuits

- Given an undirected graph, can you form a simple path containing every edge? Euler tried to answer this question in the 18th century.

Definition 4.1: Euler Path

An Euler path is a path in a connected undirected graph which includes every edge exactly once.

- When you have an Euler path that starts and finishes at the same vertex, you have an Euler circuit.

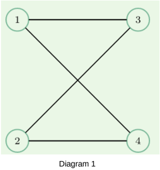
Definition 4.2: Euler Circuit

An Euler circuit is a circuit in a connected undirected graph which includes every edge exactly once.

Eulerian Paths and Circuits

Example 4.1: Finding Euler Paths

For each of the following graphs, does an Eular circuit exists?. If so give it. If not, does euler path exists? if so give it.



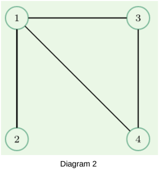


Figure 4.1: Finding Euler paths

Solution

Eulerian Paths and Circuits

Example 4.2: Finding Euler Paths

For each of the following graphs, does an Eular circuit exists?. If so give it. If not, does euler path exists? if so give it.

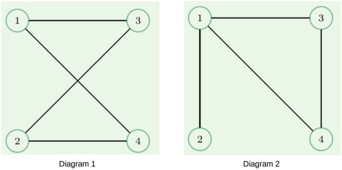


Figure 4.1: Finding Euler paths

Solution

The first graph has a Euler circuit $1, 4, 2, 3, 1$
The second graph has no Euler circuit but it has a Euler path $2, 1, 3, 4, 1$

Eulerian Paths and Circuits

Theorem 4.1

If a graph has an Euler circuit then every vertex must have even degree.

Proof.

W.l.o.g. let the Euler circuit begin at vertex u and travel along the edge $\{u, v\}$ to vertesx v This edge contributes 1 to the degree u. For each vertex traversed in the Euler circuit, the incoming edge contributes 1 to the same vertex's degree.
Finally, the circuit terminates at vertex u again, contributing 1 to the degree of u
Since the starting vertex u was chosen arbitrarily, it must be that an Euler circuit is only possible when every vertex in the graph has even degree. □

Hamilton Paths and Circuits

Definition 4.3: Hamilton Circuit

A Hamilton circuit is a simple circuit in a connected undirected graph which passes through every vertex other than the starting and ending vertex exactly once.

- Hamilton paths and circuits are much easier to find then Euler paths and circuits.

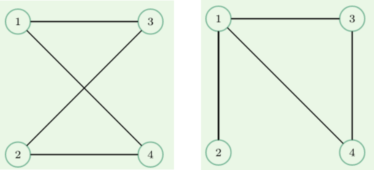


Diagram 1 Diagram 2

Figure 4.2: Finding Euler paths

- One Hamilton cycle of many for graph 1 is $1, 4, 2, 3, 1$
- Graph 2 does note have a Hamilton cycle, but it has several Hamilton paths. One is $3, 4, 1, 2$

Hamilton Paths and Circuits

- There are no known theorems giving necessary and sufficient conditions for the existence of a Hamilton circuit.
- However, some sufficient conditions are known. One is Dirac's theorem.

Theorem 4.2: Dirac's Theorem

If a simple undirected graph has order $n \geq 3$ and every vertex in the graph has degree greater than or equal to $\frac{n}{2}$, then the graph has a Hamilton circuit.

Traveling Salesperson

Example 4.3: Traveling Salesperson

- The classic problem for Hamilton circuits in computer science is the traveling salesperson problem.
- This is a classic problem in both graph theory and complexity theory. It is well-known that the traveling salesperson problem is NP-hard That is, it cannot be solved efficiently (unless $P = NP$)
- The traveling salesperson problem looks to find the minimal Hamilton path in a graph.
- What makes a Hamilton path minimal? Well, it involves a weighted graph. Something we will not study formally in this course.
- But, we can give a general idea. Imagine that every edge in an undirected graph has a "weight", "cost", or "distance" for traveling along it.
- Can you find the "minimum cost" or "minimum distance" Hamilton circuit in this graph? That is the traveling salesperson problem.

Traveling Salesperson

Example: Traveling Salesperson

- Note that edge weights have nothing to do with their "length" when drawn.
- While it is possible to draw a graph such that the edge's length is proportional to its weight, this is not done in practice because it makes the drawing quite challenging in general.
- Rather, we just put a number alongside the edge indicating its weight.
- The solution to figure 4.3 is $1, 3, 7, 9, 8, 4, 5, 2, 6, 1$

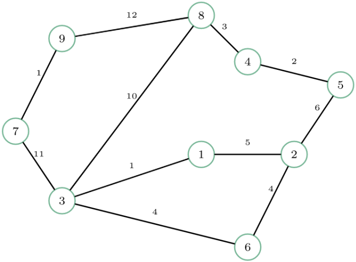


Figure 4.3: A weighted graph.

Exercises

- For each of the below graphs, does an Euler circuit exist? If so, give it. If not, does an Euler path exist? If so, give it.

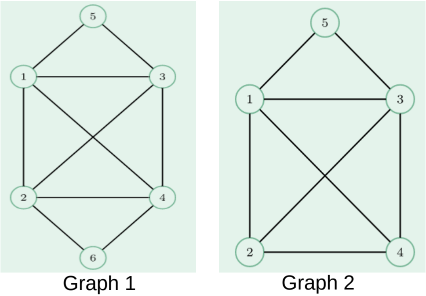


Figure 4.4: A weighted graph.

Solution

- G_1 has many Euler circuits. One is:

$$1, 5, 3, 1, 2, 6, 4, 2, 3, 4, 1$$

- G_2 does not have any Euler circuits. But, it has many Euler paths. One is:

$$2, 1, 3, 2, 4, 1, 5, 3, 4$$

- Consider a graph where vertices are people and edges connect people which are friends (or know each other). This kind of graph is often called an acquaintanceship graph.
- There is a theory in sociology that every person in the world is connected by six or fewer social connections. "Six degrees of separation".
- This can be modeled as an acquaintanceship graph. The theory is then restarted to say that every vertex in the graph is connected by a path of length at most six.

Erdős Numbers

- Consider a graph where vertices are people and edges connect people who work together or who have worked together. This is a collaborator graph.
- In academia, the collaborator graph created by having co-authorship with another researcher has some interesting properties.
- One researcher, Paul Erdős, was particularly prolific in his writings. He published at least 1500 papers
- In honour of his contribution to mathematics, people conceived the Erdős number of a researcher as the length of the path from that researcher to Erdős in the co-authorship graph.
- Erdős numbers are used both as a study of how collaborative mathematics is, as well as a small source of prestige among academics.

This is the END of

Discrete Structures II

Prepare Well for the

Examination