GRAPH THEORY

Amin Witno

Preface

Written at Philadelphia University, Jordan for Math 351, these notes[1] were used first time in the Spring 2006 semester. They have since been revised[2] and shall be revised again as often as the author teaches the course. Outline notes are more like a revision. No student is expected to fully benefit from these notes unless they have regularly attended the lectures.

1 Graphs

Definition. A *graph* G is a composite of a set V_G of *vertices* and another set E_G of *edges*, where an edge is a set of two distinct vertices. For example we may have $V_G = \{1, 2, 3, 4\}$ with $E_G = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}\}$. When there is no ambiguity we sometimes write V and E instead of V_G and E_G, respectively.

If we replace E by a multiset, allowing repetition of edges, then G is called a *multigraph*. Sometimes a multigraph is allowed to have *loops*, which are elements $aa \in E$.

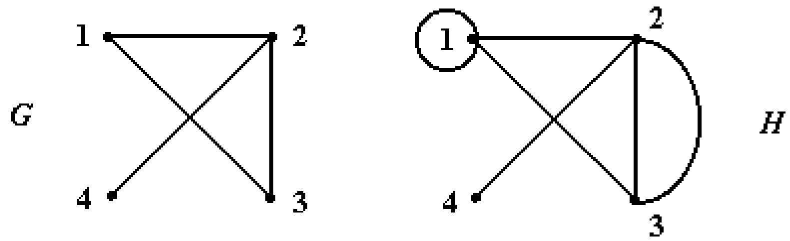


Figure 1: The graph G and a multigraph H with loop

Definition. For brevity we denote the edge $\{a, b\} \in E$ simply by ab. In particular when $ab \in E$ we say that the vertices a and b are *adjacent*. Then the *degree* of a vertex a, written $\deg(a)$, is the number of vertices of G which are adjacent to a.

[1]Copyrighted under a Creative Commons License
[2]Last Revision: 30–05–2006

Euler's Theorem In any graph or multigraph, the degree sum of all vertices is twice the number of edges. In particular the degree sum is an even number.

We shall identify four graphs which are of special characteristics.

1. A *complete* graph K_n is a graph with n vertices all of which are adjacent one to another. In particular K_3 is also called a *triangle*.

2. $K_{m,n}$ is a graph whose vertices are partitioned into two sets of m and n elements such that two vertices are adjacent if and only if they belong to different sets. A graph with this property is called *complete bipartite*.

3. A *path* P_n is a graph with vertices $\{a_1, \ldots, a_n\}$ and edges $\{a_1a_2, a_2a_3, \ldots, a_{n-1}a_n\}$. It can also be called a path from a_1 to a_n.

4. For $n \geq 3$, C_n is the graph P_n with one additional edge: a_na_1. Consequently it is called a *closed path* or a *cycle*.

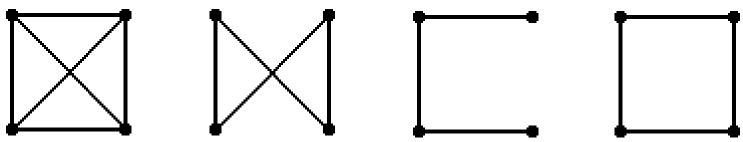


Figure 2: From left to right, the graphs $K_4, K_{2,2}, P_4, C_4$.

Exercise. Verify Euler's Theorem for the graphs $K_n, K_{m,n}, P_n, C_n$.

Definition. A graph is *d-regular* if all its vertices have degree d. For example K_3 is 2-regular.

Exercise. Analyze the regularity of the graphs $K_n, K_{m,n}, P_n, C_n$.

Definition. If $V = \{a_1, \ldots, a_n\}$ then the degree sequence of G is $\deg(a_1), \ldots, \deg(a_n)$ arranged in decreasing order. For example the degree sequence of P_5 is $2, 2, 2, 1, 1$.

Exercise. Find the degree sequences of the graphs $K_n, K_{m,n}, P_n, C_n$.

Definition. A sequence of integers $d_1 \geq d_2 \geq \ldots \geq d_n \geq 0$ is *graphical* if it is the degree sequence of some graph. For example $2, 2, 2, 1, 1$ is graphical.

Exercise. Is this a graphical sequence? Justify your answer.

1. $3, 3, 2, 2, 2$

2. $3, 2, 2, 1$

3. $4, 3, 3, 2, 1, 0$

4. $5, 3, 2, 2$

Graphical Degree Sequence Algorithm
Goal: To determine whether or not a sequence is graphical.

1. Delete the first integer, say k.
2. From what remains, subtract the first k numbers each by 1. If this is not possible then stop, the sequence is not graphical. If all terms become 0, stop, the sequence is graphical.
3. Rearrange the newly obtained sequence in decreasing order and repeat the above steps.

Exercise. Apply the algorithm to these sequences. If a sequence is graphical, try to draw a graph satisfying the given degree sequence.

1. $3, 2, 2, 1, 1, 1$
2. $3, 2, 2, 1$
3. $5, 4, 4, 3, 3, 3, 3, 2, 2, 1$
4. $4, 3, 3, 2, 1, 0$
5. $2, 1, 1, 0$
6. $5, 3, 2, 2, 1, 1$

Theorem For multigraphs, the sequence $d_1 \geq d_2 \geq \ldots \geq d_n \geq 0$ is graphical if and only if the sum is an even number.

2 Isomorphisms

Definition. Two graphs are *isomorphic* to each other, written $G \simeq H$, if there is a bijection $f : V_G \to V_H$ such that $a, b \in V_G$ are adjacent if and only if $f(a), f(b) \in V_H$ are adjacent. For example $K_3 \simeq C_3$ and both are called triangles. Also $K_2 \simeq K_{1,1} \simeq P_2$.

Exercise. Find more isomorphisms between the graphs $K_n, K_{m,n}, P_n, C_n$.

Remark. Note that if $G \simeq H$ then $|V_G| = |V_H|$, $|E_G| = |E_H|$, and their degree sequences must be identical. However none of these is a sufficient condition for isomorphism.

Exercise. Find k non-isomorphic graphs with the given degree sequence.

1. $3, 2, 2, 1, 1, 1 \quad (k = 3)$
2. $4, 4, 3, 2, 2, 1 \quad (k = 2)$
3. $5, 3, 2, 2, 1, 1, 1, 1 \quad (k = 3)$
4. $2, 2, 2, 2, 2, 2, 1, 1 \quad (k = 4)$
5. $3, 3, 3, 3, 3, 3, 3, 3, 3 \quad (k = 2)$

Definition. A graph G is a *subgraph* of another graph H if $V_G \subseteq V_H$ and $E_G \subseteq E_H$. We denote this relation by $G \subseteq H$ and say that H *contains* G. For example $P_3 \subseteq C_3$ and $K_{2,2} \subseteq K_{2,4}$.

Exercise. Find more examples of subgraphs among the graphs $K_n, K_{m,n}, P_n, C_n$.

Definition. A graph G is *connected* if there is a path from any vertex to any other vertex in G. A *component* of G is a maximal connected subgraph of G. A graph is *disconnected* if it is not connected, or equivalently, if it has more than one component. It is clear that isomorphic graphs must have the same number of components.

Definition. The *complement* of a graph G is the graph $\bar{G}$ such that $V_{\bar{G}} = V_G$ and $ab \in E_{\bar{G}}$ if and only if $ab \notin E_G$.

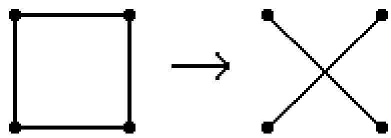


Figure 3: The graph C_4 and its complement.

Exercise. Draw the complements of $P_4, K_5, K_{3,3}$.

Proposition If G is disconnected then $\bar{G}$ is connected.

Definition. A graph G is *self-complementary* if $\bar{G} \simeq G$. For example P_4. It follows from the proposition that a self-complementary graph must be connected.

Exercise. Find two self-complementary graphs with 5 vertices.

Proposition If G is self-complementary then $|V_G| \bmod 4 = 0$ or 1 and $|E_G| = \frac{n(n-1)}{4}$.

Exercise. Is there a self-complementray graph with 3 vertices? 8? 15? 21?

3 Trees

Definition. A *tree* is a connected graph which contains no cycles. For example P_4. In general a graph which contains no cycles is called *acyclic*.

Exercise. Which ones of the graphs $K_n, K_{m,n}, P_n, C_n$ are trees?

Theorem Let G be a connected graph. The following are equivalent.

1. G is acyclic.
2. $|V| = |E| + 1$
3. Every edge of G is a *bridge*, meaning that if removed, the graph would become disconnected.
4. There is a unique path between any two vertices of G.

Proposition A tree contains a vertex of degree 1, which is called a *leaf*.

Exercise. Prove that adding an edge to a tree will produce a cycle.

Theorem The degree sequence of a tree is of the form $d_1 \geq d_2 \geq \ldots \geq d_n = 1$ with $\sum d_i = 2n - 2$. Conversely any sequence with this property is the degree sequence of a tree.

Exercise. Draw all possible trees with 4, 5, or 6 vertices.

Proposition Let n_i denote the number of vertices of degree i in a graph G. Then the number of leaves in G is $n_1 = 2 + \sum_{i\geq 3}(i-2)n_i$.

Definition. A *spanning tree* of a graph G is a tree $T \subseteq G$ such that $V_T = V_G$. For example P_4 is a spanning tree of both C_4 and K_4.

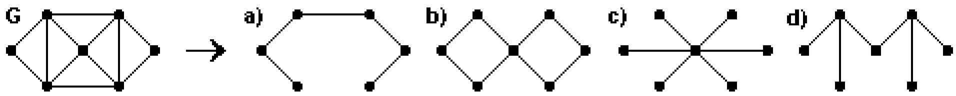


Figure 4: Only (d) is a spanning tree of the graph G.

Depth-First Search Algorithm
Goal: To determine whether or not a given graph G is connected, in which case it produces a spanning tree of G.

1. We shall label the vertices of G as $1, 2, \ldots$ based upon the order of traversal as follow.
2. Start with an arbitrary vertex 1.
3. Move to any vertex adjacent to the current selection which have not been labeled. If no such vertex exists backtrack to a previously visited vertex.
4. Repeat until no more move is possible.
5. If all vertices of G are labeled then G is connected and this traversal generates a spanning tree of G.

Kirchoff's Algorithm
Goal: To count the number of spanning trees of a connected labeled graph G.

1. Assume that $V = \{v_1, v_2, \ldots, v_n\}$.
2. Let M be an $n \times n$ matrix given by

$$(M)_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \text{ or else} \\ -1 & \text{if } ij \in E \\ 0 & \text{if } ij \notin E \end{cases}$$

3. Compute any cofactor of M. This is the number of spanning trees of G.

Exercise. Apply Kirchoff's Algorithm to the graphs $K_4, K_{2,3}$, and C_5.

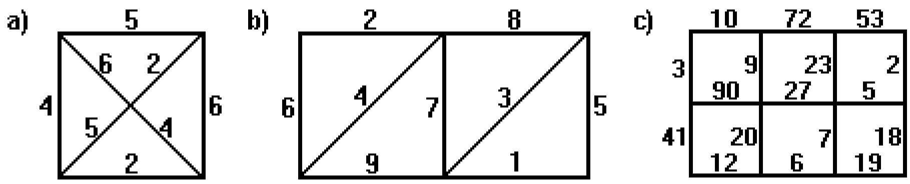
Corollary The number of labeled trees with $n \geq 2$ vertices is the number of spanning trees of a labeled K_n, which is n^{n-2}.

Definition. A graph is *weighted* if every edge is associated with a numerical value called the edge's *weight*. For us, weighted graphs are allowed only non-negative values. A *minimal spanning tree* of a weighted graph is a spanning tree with least total weight.

Kruskal's Algorithm
Goal: To produce a minimal spanning tree of a weighted graph G.

1. Select an edge of least weight.
2. Add the next least-weight edge, as long as this action does not create a cycle.
3. Repeat until $|E| = |V| - 1$.

Exercise. Apply Kruskal's Algorithm to find a minimal spanning tree.



Definition. The *neighborhood* of $v \in V$ is the set $N(v) = \{w \in V \mid vw \in E\}$. If $S \subseteq V$ then the neighborhood of S is the set $N(S) = \bigcup\{N(v) \mid v \in S\}$.

Prim's Algorithm
Goal: To produce a minimal spanning tree of a weighted graph G.

1. Select an arbitrary vertex. Suppose S is the set of vertices which have already been selected.
2. Add to S one more vertex from $N(S)$ such that the corresponding edge is of least weight.
3. Repeat until $|S| = |V|$.

Exercise. Repeat the previous exercise using Prim's Algorithm.

4 Bipartite Graphs

Definition. A *bipartite graph* is a graph with $V = X \sqcup Y$, that is disjoint union of two sets, such that every edge connects only vertices from opposite sets. For example $K_{m,n}$ are all bipartite. Note that if G is disconnected then G is bipartite if and only if each component is bipartite, hence we may well assume that G is connected throughout this chapter.

Bipartite Graph Coloring Algorithm
Goal: To determine whether or not G is bipartite, and produce the bipartition sets X and Y such that $V = X \sqcup Y$.

1. Select an arbitrary vertex and color it black. Suppose S is the set of vertices which have already been colored.
2. For each vertex in $N(S)$ color it white or black such that adjacent vertices have distinct colors. If this is not possible then G is not bipartite.
3. Repeat until $|S| = |V|$. In the end $V = X \sqcup Y$ with X, Y the black and white vertices, respectively.

Theorem A graph is bipartite if and only if it contains no cycle of odd vertices. In particular a tree is bipartite.

Exercise. Which of the graphs $K_n, K_{m,n}, P_n, C_n$ are bipartite?

Proposition Suppose $|V| = n$. If G is bipartite then $|E| \leq \frac{n^2}{4}$, where equality holds if and only if $G \simeq K_{\frac{n}{2},\frac{n}{2}}$.

Definition. A *matching* in a bipartite graph G is a subgraph $M \subseteq G$ in which the edges are mutually disjoint. If $V_G = X \sqcup Y$ then a matching M is called *complete* if for each $v \in X$ there is $vw \in E_M$. And if, in addition, $|X| = |Y|$ then a complete matching is called *perfect*.

Hall's Marriage Theorem Let G be bipartite with $V = X \sqcup Y$. Then G has a complete (and perfect if $|X| = |Y|$) if and only if $|S| \leq |N(S)|$ for every $S \subseteq X$.

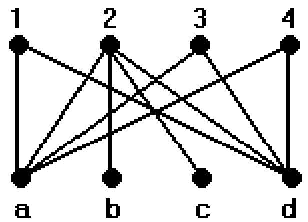


Figure 5: A complete matching is not possible since $|S = \{1, 3, 4\}| > |N(S) = \{a, d\}|$.

Proposition Suppose $V = X \sqcup Y$. If G is regular then $|X| = |Y|$ and G has a perfect matching.

Exercise. Draw all regular bipartite graphs with 6 vertices.

5 Coloring

Definition. The *chromatic number* $\chi(G)$ of a graph G is the least number of colors needed to color the vertices of G such that adjacent vertices have distinct colors. For example $\chi(G) = 2$ if and only if G is bipartite. Note that if G is disconnected then $\chi(G)$ is simply the largest chromatic number of a component of G, hence we may well assume that G is connected throughout this chapter.

Exercise. Find the chromatic numbers of the graphs $K_n, K_{m,n}, P_n, C_n$.

Proposition Results concerning chromatic numbers

1. If $G \subseteq H$ then $\chi(G) \leq \chi(H)$.
2. If $\chi(G) = k$ then G contains k vertices each of which has degree at least $k-1$.
3. $\chi(G) \leq \Delta(G) + 1$ where $\Delta(G)$ is the largest degree in V_G.
4. In (2) equality holds if and only if $G \simeq K_n$ or C_n with n odd, else $\chi(G) \leq \Delta(G)$.

Sequential Coloring Algorithm
Goal: To color the pre-ordered vertices of G.

1. Assume the vertices of G are ordered $v_1, \ldots, v_n$.
2. From $i = 1$ to n, assign to v_i the smallest possible positive integer such that adjacent vertices have distinct integers.
3. In the end the integers represent the different colors used.

Definition. The *color-degree* of a vertex is the number of distinct colors of vertices which are adjacent to it.

Maximum Color-Degree Coloring Algorithm
Goal: To color the vertices of G.

1. Select a vertex with largest degree and assign to it the color (integer) 1. Suppose S is the set of vertices which have already been colored.
2. Select a vertex in $N(S)$ with largest color-degree. If it is not unique, choose the one with the largest degree. Color this vertex with the smallest possible positive integer.
3. Repeat until all vertices have been colored.

6 Planar Graphs

Definition. A graph is *planar* if it can be drawn in the plane such that no edges are crossing each other. This particular drawing of the graph is called a *plane graph.* For example K_4 is planar but $K_{3,3}$ is not. Note that if G is disconnected then G is planar if and only if each component is planar, hence we may well assume that G is connected throughout this chapter.

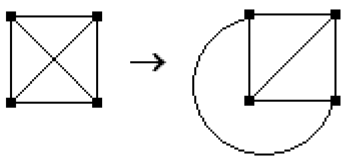


Figure 6: The graph K_4 drawn as a plane graph without edge crossing.

Proposition If $G \subseteq H$ and G is not planar then neither is H. In particular $K_{m,n}$ is not planar if $m, n \geq 3$.

Definition. A plane graph partitions the plane into subsets called *regions.* For example the plane graph of K_4 has 4 regions, one of which is exterior to the graph.

Euler's Formula Suppose G is a plane graph, or multigraph, with v vertices, e edges, and r regions. Then $v + r = e + 2$.

Corollary

1. If G is planar with $|V| \geq 3$ then $|E| \leq 3|V| - 6$. In particular K_5 is not planar and neither are K_n for all $n \geq 6$.

2. If G is planar then there is a vertex of degree 5 or less.

3. If G is planar and contains no triangles then $|E| \leq 2|V| - 4$.

Definition. Two graphs are *homeomorphic* if one can be obtained from the other by adding vertices along some edges. For example any two cylces are homeomorphic.

Kuratowski's Theorem A graph is planar if and only if it contains no subgraph homeomorphic to K_5 or $K_{3,3}$.

Definition. The *dual graph* G' of a plane graph G is a graph whose vertices are the interior regions of G and $r_i r_j \in E_{G'}$ if and only if the regions r_i and r_j are *neighbors*, meaning that they share and edge in G. For example $K_4' = K_3$.

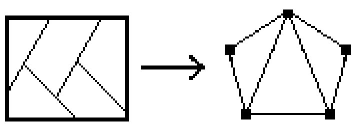
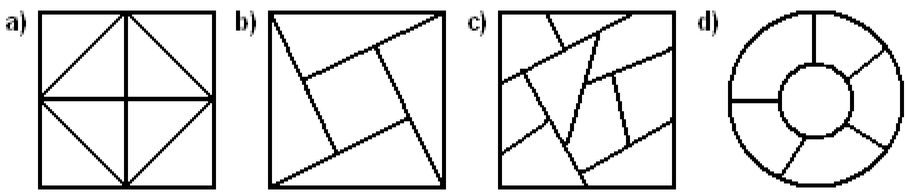


Figure 7: A plane graph and its dual graph.

Exercise. For each plane graph, draw the dual graph.



Definition. The *chromatic number* of a plane graph G is the least number of colors needed to color the interior regions of G such that neighboring regions have distinct colors. It is clear that the chromatic number of a plane graph equals $\chi(G')$.

Exercise. Find the chromatic numbers of the plane graphs from the previous exercise. Then redo each problem, this time by coloring the dual graphs.

The Four-Color Theorem If G is planar then $\chi(G) \leq 4$. Equivalently, the chromatic number of a plane graph is at most 4.

7 Walks

Definition. A *walk* is a sequence of edges $v_1v_2, v_2v_3, \ldots, v_{n-1}v_n$ which are not necessarily distinct (unless the walk is a path). In this case we say that the walk is from v_1 to v_n of *length* $n-1$. The *distance* between two vertices a and b, written $d(a,b)$, is the length of the shortest walk from a to b, if it exists, otherwise let $d(a,b)=\infty$. Note that the shortest walk is necessarily a path. Furthermore, in a weighted graph, $d(a,b)$ is understood to be the minimum total weights of all possible walks from a to b.

Definition. The *diameter* of a graph G is given by $\operatorname{diam}(G)=\max\{d(a,b) \mid a,b\in V\}$. For example $\operatorname{diam}(G)=1$ if and only if G is complete. Note also that $\operatorname{diam}(G)=\infty$ if and only if G is disconnected.

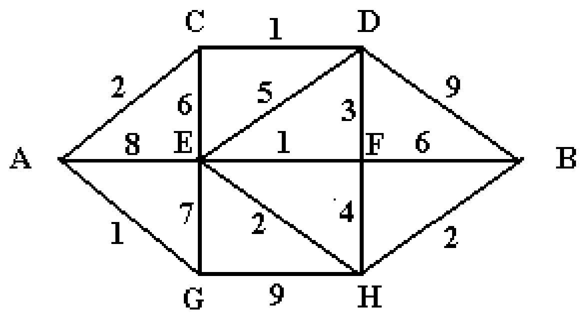
Exercise. Find the diameters of the graphs $K_n, K_{m,n}, P_n, C_n$.

Proposition If $\operatorname{diam}(G)\geq 3$ then $\operatorname{diam}(\bar{G})\leq 3$. Hence if G is self-complementary then $\operatorname{diam}(G)\leq 3$.

Dijkstra's Algorithm
Goal: To compute $d(a,b)$ in a weighted graph and find the shortest path from a to b.

1. Label the vertex a by the pair $(-,0)$. Suppose S is the set of vertices v which have already been labeled by $(-,w_v)$.
2. For each vertex $n\in N(S)$, say with edge ns of weight w for some vertex $s\in S$, calculate the number $w_n = w + w_s$. Choose the least possible of such number and label the corresponding vertex n by (s,w_n).
3. Repeat until b is labeled, say $(-,w_b)$, in which case $d(a,b)=w_b$ and the shortest path from a to b can be found by backtracking the first entry in the labels.

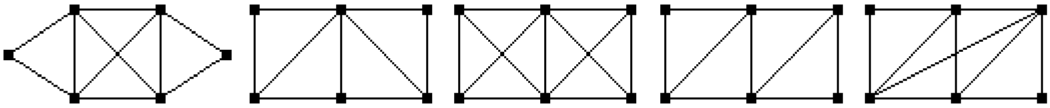
Exercise. Apply Dijkstra's Algorithm to find $d(A,B)$.



Definition. An *Euler walk* in a connected graph G is a walk through all the edges of G without repeating any of them. If an Euler walk is closed, meaning that it starts and ends at the same vertex, then it is called an *Euler circuit*.

Theorem A connected graph, or multigraph, has an Euler walk from a to b if and only if $\deg(a)$ and $\deg(b)$ are odd while all the other vertices have even degrees. The graph, or multigraph, has an Euler circuit if and only if all vertices have even degrees.

Exercise. Illustrate the theorem on the following graphs.

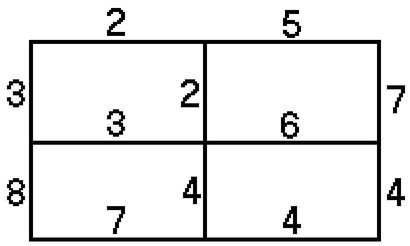


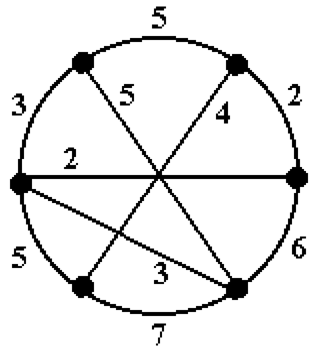
The *Chinese Postman Problem* asks for the shortest closed walk going through every edge in a graph, possibly weighted. If any, an Euler circuit would give the best solution, or else such a walk would have to repeat one or more edges.

Chinese Postman Algorithm
Goal: To find the shortest closed walk going through each edge of a (weighted) graph.

1. Identify all vertices which have odd degrees. By Euler's Theorem their number is even.
2. Pair up these odd vertices $\{a_1, b_1\}, \ldots, \{a_n, b_n\}$ in such a way that $\sum d(a_i, b_i)$ is the least possible.
3. The shortest walk is through all the edges of G plus the shortest paths from a_i to b_i found above.

Exercise. Solve the Chinese Postman Problem, without and then with the weights.





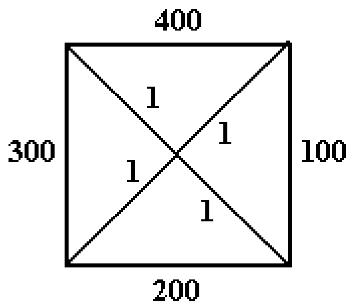
Definition. A *Hamilton cycle* in a graph G is a cycle $C_n \subseteq G$ where $n = |V_G|$. In other words, a Hamilton cycle is a closed walk through all vertices of G without repeating any of them (except of course the last vertex). The graph G is *Hamiltonian* if it contains a Hamilton cycle. For example, by definition, a tree is not Hamiltonian. It is also clear that a Hamiltonian graph must be connected.

Proposition Results concerning Hamiltonian graphs

1. If a graph contains a leaf then it is not Hamiltonian.
2. If a graph contains a bridge then it is not Hamiltonian.
3. If removing a vertex disconnects the graph then it is not Hamiltonian.
4. If a vertex has degree 2 then any Hamilton cycle must contain both its edges.
5. If every vertex has degree at least $\frac{|V|}{2}$ then the graph is Hamiltonian.
6. If $\deg(a) + \deg(b) \geq |V|$ for all $ab \notin E$ then the graph is Hamiltonian.

The *Traveling Salesman Problem* asks for the shortest closed walk going through every vertex in a weighted graph. One way to find such a walk is to try out all possible Hamilton cycles, if any, and select the one with least total weight. However, there is no guarantee that a Hamilton cycle will give the best solution and in general we have no efficient algorithm for solving the Traveling Salesman Problem.

Exercise. Solve the Traveling Salesman Problem for the following weighted graph.



8 Matrices

Definition. Suppose $V = \{v_1, \ldots, v_n\}$. The *adjacency matrix* of G is the $n \times n$ matrix A given by $(A)_{ij} = 1$ if $ij \in E$, and 0 otherwise. For example the adjacency matrix of C_4 is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

Exercise. Find the adjacency matrices of the graphs $K_n, K_{m,n}, P_n, C_n$.

Proposition Let A denote the adjacency matrix of a graph G. Then $(A^n)_{ij}$ represents the number of walks of length n from i to j. In particular $(A^2)_{ii} = \deg(v_i)$.

Corollary The number of triangles contained in the graph is given by $\frac{1}{6}\sum_{i\geq 1}(A^3)_{ii}$.

Exercise. Find the number of triangles contained in K_4 and K_5.

Definition. A *permutation matrix* is a square matrix obtained from the identity by permuting its rows. For example $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. Note that a permutation matrix is a special case of an orthogonal matrix, satisfying $P^{-1} = P^T$.

Theorem Suppose A and B are the adjacency matrices of the graphs G and H, respectively. Then $G \simeq H$ if and only if $B = PAP^T$ for some permutation matrix P.

Definition. Suppose that $V\{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Then the *incidence matrix* of G is the $n \times m$ matrix I given by $(I)_{ij} = 1$ if $v_i \in e_j$, and 0 otherwise. For example the incidence matrix of P_4 can be given by

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Exercise. Find the incidence matrices of the graphs $K_n, K_{m,n}, P_n, C_n$.

Exercise. Convert the following adjacency matrices to incidence matrices.

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

Exercise. Convert the following incidence matrices to adjacency matrices.

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

Definition. Suppose $V = \{v_1, \ldots, v_n\}$. The *distance matrix* of G is the $n \times n$ matrix D given by $(D)_{ij} = d(v_i, v_j)$. For example the distance matrix of P_4 is given by

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}.$$

Exercise. Find the distance matrices of the graphs $K_n, K_{m,n}, P_n, C_n$.

Exercise. In the previous exercises, convert the adjacency matrices to distance matrices, and the incidence matrices to distance matrices.

From A to D Algorithm
Goal: To obtain the distance matrix D from a given adjacency matrix A.

1. Compute the matrices $I = A^0, A = A^1, A^2, \ldots, A^n$, where $n = |V|$.
2. $(D)_{ij}$ is the smallest k such that $(A^k)_{ij} \neq 0$, or ∞ if no such k exists.

Floyd-Warshall Algorithm
Goal: To find the distance matrix D of a given weighted graph G.

1. Assume that $V = \{v_1, \ldots, v_n\}$.

2. Let $(D)_{ij} = \begin{cases} 0 & \text{if } i = j \text{ or else} \\ w & \text{if } v_iv_j \in E \text{ with weight } w \\ \infty & \text{if } v_iv_j \notin E \end{cases}$

3. From $k = 1$ to n do the following step.

4. For each element of D, replace $(D)_{ij}$ by $(D)_{ik} + (D)_{kj}$ if this sum is smaller.

9 Digraphs

Definition. A *digraph* G^* is a composite of a set V_G of *vertices* and another set E_G^* of *arcs*, where an arc is an ordered pair of two distinct vertices. For example we may have $V = \{1, 2, 3, 4\}$ with $E^* = \{(1, 2), (1, 3), (2, 3), (3, 2), (4, 2)\}$. As with graphs, for brevity we denote the arc $(a, b) \in E^*$ simply by ab, in which case we say that the arc ab is *oriented* or *directed* from a to b. In contrast to graphs, however, here $ab \neq ba$. The degree of a vertex is therefore classified as an *indegree* or an *outdegree* depending whether an arc is directed into or away from the vertex, respectively.

Euler's Theorem for Digraphs In any digraph the indegree sum, the outdegree sum, and the number of arcs are all equal.

Remark. Most of the results on graphs in the preceeding chapters can be modified appropriately and applied for digraphs. In particular every walk in a digraph must obey the direction (orientation) of each arc.

Exercise. Apply Dijkstra's Algorithm to the following weighted digraph, start at A.

Definition. A digraph is *strongly connected* if there is a walk from every vertex to any other vertex. To be strongly connected it is necessary that the graph (that is the digraph without the orientation) is connected, however this condition is not sufficient.

Theorem A strongly connected digraph has an Euler circuit if and only if every vertex has equal indegree and outdegree. Or there is an Euler walk from a to b if and only if a has one more outdegree than indegree, b has one more indegree than outdegree, and every other vertex has equal indegree and outdegree.

Theorem Suppose G^* is a strongly connected digraph. If every vertex has total degree (in plus out) at least $|V|$ then G^* has a Hamilton cycle.

Proposition Every connected graph can be made a strongly connected digraph by orienting each edge in a one-way direction, provided that the graph contains no bridges. In general a digraph with one-way orientation for each arc is called *anti-symmetric*.

One-Way Street Algorithm
Goal: To turn a connected graph G into a strongly connected anti-symmetric digraph.

1. Find a labeled spanning tree $T \subseteq G$ using Depth-First Search Algorithm, say $V_T = \{1, 2, \ldots, n\}$.

2. For each edge $ij \in E_G$ with $i < j$ we orient it from i to j if $ij \in E_T$, and otherwise from j to i.

3. If G has no bridges, the resulting G^* is strongly connected.

Definition. A *tournament* is an anti-symmetric digraph such that for every two distinct vertices a and b, either ab or ba is an arc. In other words, a tournament is a complete graph with a one-way orientation for each edge. In applications, a tournaments may represents round-robin matches between teams, and $ab \in E^*$ means a defeated b. In this case the best team should be one with the largest outdegree.

Theorem Every tournament G^* has a Hamilton path, that is a sub-digraph P_n with $n = |V|$. Furthermore, the following statements are all equivalent.

1. G^* is *transitive*, meaning that if ab and bc are arcs then so is ac.
2. G^* has a unique Hamilton path.
3. Every vertex has a different outdegree.

Corollary Suppose a_1 is a vertex with largest outdegree in a tournament. Then $d(a_1, b) \leq 2$ for any other vertex b.

Definition. A *canonical ordering* of a digraph is a labeling of its vertices $v_0, \ldots, v_n$ in such a way that ij is an arc only if $i < j$. With a particular canonical ordering we call the vertex v_0 the *root* of the digraph.

Theorem A digraph has a canonical ordering if and only if it is acyclic, which means as with graphs, that it contains no cycles.

Bellman's Algorithm
Goal: To find the shortest paths, if any, from the root v_0 to every other vertex in a weighted digraph with canonical ordering $v_0, \ldots, v_n$.

1. Set $d_0 = 0,\ d_1 = \cdots = d_n = \infty,\ p_0 = \cdots = p_n = -1$.
2. For $k = 1$ to n do the following step.
3. Let $d_k = \min\{d_j + w(v_j, v_k) \mid j = 0, \ldots, k-1\}$ and let $p_k = j$ which gives this minimum.
4. In the end, $d_k = d(v_0, v_k)$ and the shortest path is obtained by backtracking the values $p_k = j, p_j = i, p_i = \cdots = 0$ corresponding to $v_k, v_j, v_i, \ldots, v_0$.

Definition. A digraph is a *directed tree* or *rooted tree* if it is a tree (ignoring orientation) and if there is a unique vertex, called the *root*, with indegree 0.

Proposition In a directed tree there is a unique path from the root to every other vertex. Moreover, every vertex other than the root has indegree 1.

To Learn More

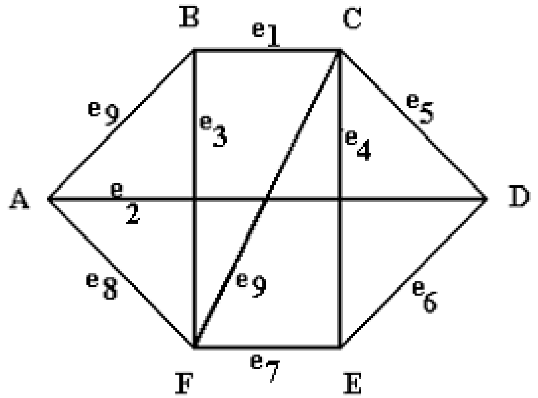
Any one of the following textbooks can be used to teach yourself Graph Theory.

1. Goodaire and Parmenter, *Discrete Mathematics with Graph Theory*, Third Edition 2006, Prentice Hall

2. Bondy and Murty, *Graph Theory with Applications*, Revised Edition 2005, Wiley
3. Chartrand and Zhang, *Introduction to Graph Theory*, 2005, McGraw Hill
4. Buckley and Lewinter, *A Friendly Introduction to Graph Theory*, 2003, Prentice Hall

Project

Problem A



In the given graph G first you must assign the weights $e_1, e_2, \ldots, e_9$, replacing them by the nine digits of your University Number from left to right, respectively. Note that there are two edges labeled e_9. Then you may do the following assignments.

1. Solve the Chinese Postman Problem for the weighted graph G.
2. Apply Dijkstra's Algorithm to find the shortest distances from A to all the other vertices in G.
3. Apply Floyd-Warshall Algorithm to find the shortest distances between every two vertices of G.
4. (Bonus) Solve the Traveling Salesman Problem for the weighted graph G, but do this *only* if you are able to prove that your solution is correct.

Problem B

Let G be the graph whose vertices are the digits $0, 1, \ldots, 9$ and two distinct vertices are adjacent if and only if both digits appear in one of the following numbers.

$$30 \qquad 189 \qquad 287 \qquad 528 \qquad 641 \qquad 936 \qquad N$$

where N is the number from the last four digits of your University Number. For example if your University Number is 200316568 then $N = 6568$. First you must draw the graph G and then do the following assignments.

1. Apply the Sequential Coloring Algorithm to color the vertices of G, from 0 to 9.
2. Apply the Maximum Color-Degree Algorithm to color the vertices of G.
3. (Bonus) Find $\chi(G)$, but do this only if you are able to prove that your answer is correct.