1 Introduction

1.1 Definitions

- A *graph* consists of a finite non-empty set V of *vertices* and a finite collection E of unordered pairs of elements of V called *edges*.
- A vertex x is *adjacent* to y if $\{x, y\}$ is an edge.
- The edge $\{x, y\}$ is *incident* with x and y.
- An *isolated vertex* is a vertex with no edges incident.
- A *simple graph* is a graph that contains no multiple edges or loops.
- A graph is a *multigraph* if it has no loops but a pair of vertices may have more than one edge joining them.
- A graph is a *pseudograph* if loops and multiple edges are permitted.
- The *degree* of a vertex v, $\deg(v)$, is the number of edges incident with v. If v is the vertex of a simple graph, then $0 \leq \deg(v) \leq n - 1$.
- The *degree sequence* of a graph is a sequence of the degrees of all the vertices in ascending order.
- A graph is said to be *regular of degree* r if the degree of each vertex is r.
- A regular graph of degree 3 is *cubic*
- An *Euler* is an ugly man on page 5.

1.2 Hand-shaking Lemma

1.2.1 Lemma

$$\sum_{u \in V} \deg u = 2|E|$$

1.2.2 Corollaries

- In any graph, the number of vertices of odd degree is even.
- Every cubic graph has an even number of vertices.

1.3 Isomorphism

1.3.1 Definition

$G_1 \simeq G_2$ if there exists a bijection from V_1 to V_2 such that v and w are adjacent in G_1 if and only if $f(v)$ and $f(w)$ are adjacent in G_2 for all $v, w \in V_1$.

1.3.2 Characterisation of Isomorphism

If $G_1 \simeq G_2$, then

- G_1 and G_2 have the same number of vertices and the same number of edges.
- G_1 and G_2 have the same degree sequence.

1.3.3 Equivalence Relation

Isomorphism is an equivalence relation with the following properties:

- *Reflexive:* Any graph is isomorphic to itself.
- *Symmetric:* For any graphs G_1 and G_2, $G_1 \simeq G_2 \Rightarrow G_2 \simeq G_1$.
- *Transitive:* For any graphs G_1, G_2 and G_3, if $G_1 \simeq G_2$ and $G_2 \simeq G_3$, then $G_1 \simeq G_3$.

1.4 Subgraphs

1.4.1 Definition

Let $G = (V, E)$ and $H = (U, F)$. Then H is a subgraph of G if $\emptyset \neq U \subseteq V$ and $F \subseteq E$.

1.5 Special Graphs

1.5.1 Null Graphs

A *null graph* is a graph with no edges, and the null graph on n vertices is denoted by $N_n = (V, E)$, where $V = \{x_1, \ldots, x_n\}$ and $E = \{\}$. Null graphs are *not interesting* (Palmer, 2005).

1.5.2 Complete Graphs

Let $n \in \mathbb{Z}^+$. K_n, the complete graph on n vertices, is a simple graph with n vertices, whose edge collection contains exactly one edge for each pair of distinct vertices.

1.5.3 Cycle Graphs

The cycle graph on n vertices is given by $C_n = (V, E)$, where $V = \{x_1, x_2, \ldots, x_n\}$ and $E = \{x_1x_2, x_2x_3, \ldots, x_nx_1\}$. Note that $C_3 = K_3$.

1.5.4 Peterson Graph

(See page 41)

1.5.5 Path Graph

The path graph on n vertices, P_n, is the graph obtained by removing an edge from C_n.

1.5.6 Wheel

A wheel on n vertices, W_n, is C_{c-1} with an additional vertex which is joined to each of the vertices in C_{n-1}.

1.6 Complement

1.6.1 Definition

The complement, $\bar{G}$, of a simple graph $G = (V, E)$, is the graph with vertex set V such that vertices are adjacent if and only if they are not adjacent in G.

2 Walks, Trails, Paths and Cycles

2.1 Definitions

- Given a graph $G = (V, E)$ with $V = \{v_0, \ldots, v_n\}$, a *walk* is a finite sequence of edges $v_0 \to v_1 \to \cdots \to v_{m-1} \to v_m$.
- The number of edges in a walk is called its *length*.
- A *trail* is a walk in which all edges are distinct (but not necessarily for edges).
- A *path* is a trail such that all vertices are distinct (except perhaps for the initial and final vertices).
- If a walk, path or trail is *closed* then the initial vertex is the same as the final vertex.
- A *cycle* is a closed path with more than one edge.
- A *triangle* is a cycle of length 3.

2.2 Characterisation of Walks, Trails, Paths and Cycles

- A graph in which the degree of each vertex is at least 2 contains a cycle.

2.3 Connectedness

A graph is *connected* if and only if every pair of vertices are joined by a path. Otherwise, the graph is *disconnected*. Any disconnected graph G can be expressed as the union of connected subgraphs, each of which is called a *component* G.

2.4 Bridges

A *bridge* is an edge of the graph, which if removed, disconnects the graph. An edge e of a connected simple graph G is a bridge of G if and only if e lies on no cycle of G.

2.5 Distance

Let G be a simple graph. Given vertices x and y, the distance $d(x, y)$ is the minimum length of an x-y path. If there is no x-y path, then $d(x, y) = \infty$.

3 Eulerian Graphs

3.1 Definitions

- An *Eulerian trail* is a closed trail that contains every vertex and every edge. That is, each edge appears precisely once, but vertices may be repeated.
- An *Eulerian graph* is a graph that contains a Eulerian trail.
- A *semi-Eulerian trail* is a trail that contains every vertex and every edge, but which is not a closed trail.
- A *semi-Eulerian graph* is a graph that contains a semi-Eulerian trail.

3.2 Characterisation of Eulerian Graphs

- An Eulerian graph G with at least 2 vertices is connected and the degree of each vertex is even.
- If G is a connected graph and the degree of each vertex of G is even, then G is Eulerian.
- A connected graph is semi-Eulerian if and only if it has exactly 2 vertices of odd degree.

3.3 Algorithms for Finding Eulerian Trails

3.3.1 Construction

1. Find a cycle C.
2. Remove it.
3. Find Eulerian trails in the remaining subgraph.
4. Restore C.
5. Repeat.

(See page 61 for an illustration)

3.3.2 Fleury's Algorithm

1. Start at any vertex, and begin to choose edges in an arbitrary manner, erasing edges and resultant isolated vertices.
2. At any vertex, choose a bridge only if there is no alternative.

(See page 64 for an illustration)

4 Hamiltonian Graphs

4.1 Definitions

- A *Hamiltonian cycle* in a graph G is a cycle that contains every vertex of G.
- A *Hamiltonian graph* is a graph that contains a Hamiltonian cycle.
- A *Hamiltonian path* in a graph G is a path that contains every vertex of G.
- A *semi-Hamiltonian graph* is a graph that is not a Hamiltonian graph, but contains a Hamiltonian path.

4.2 Characterisation of Hamiltonian Graphs

- To show that a graph is Hamiltonian, we show look for a Hamiltonian graph.
- If a graph G has a non-trivial Hamiltonian cycle, then G has a subgraph H with the following properties:
  1. H contains every vertex of G.
  2. H is connected.
  3. H has the same number of edges as vertices.
  4. Every vertex of H has degree 2.

4.3 K_n is Hamiltonian

There are $\frac{(n-1)!}{2}$ essentially different Hamiltonian cycles in K_n.

4.4 Decomposition of K_n into disjoint Hamiltonian cycles

K_n, n odd, $n \geq 3$, can be deconstructed into $\frac{n-1}{2}$ disjoint Hamiltonian cycles:

1. Place v_1 into the centre.
2. Place $v_2, v_3, v_5, \ldots, v_n, v_{n-1}, \ldots, v_6, v_4$ into the circle centred at v_1.
3. Then a Hamiltonian cycle is $v_1 \to v_2 \to v_3 \to \cdots \to v_n \to v_1$.
4. Then rotate the vertices anticlockwise by $\frac{2\pi}{n-1}$ radians, keeping the "edge-lines" fixed.

(See page 86 for an illustration)

4.5 Theorems

- *Ore's Theorem:* Let G be a simple graph with n vertices. Suppose that, for any two non-adjacent vertices x and y in G,
  $$\deg(x) + \deg(y) \geq n$$
  Then G is Hamiltonian.
- *Dirac's Theorem:* A simple graph with $n \geq 3$ vertices is Hamiltonian if $\deg(v) \geq \frac{n}{2}$ for each vertex v.
- Suppose $G = (V_1 \cup V_2, E)$ is bipartite and $|V_1| \neq |V_2|$. Then G is not Hamiltonian.
- If G is a Hamiltonian graph, then for every non-empty set $S \subseteq V$ of G, the number of components in $G - S \leq |S|$.
- Q_1 has a Hamiltonian path, and for $n \geq 2$, Q_n is Hamiltonian.

5 Bipartite Graphs

5.1 Definitions

- A graph $G = (V, X)$ is *bipartite* if the vertex-set V can be split into 2 disjoint sets V_1 and V_2 such that each edge of G joins a vertex of V_1 to a vertex of V_2. There are no edges in G joining vertices in V_1 or V_2.
- A *complete bipartite graph* $G = (V_1 \cup V_2, E)$ is a bipartite graph such that each vertex of V_1 is joined to each vertex of V_2 by exactly one edge. $K_{m,n}$ is the complete bipartite graph with $|V_1| = m, |V_2| = n$.

5.2 Characterisation of Bipartite Graphs

- A simple graph is bipartite if and only if it does not contain an odd cycle.

5.3 k-cube graph, Q_k

5.3.1 Definition

The k-cube graph, Q_k, is a regular bipartite graph graph where each vertex consists of a sequence $(a_1, a_2, \ldots, a_k)$ where each $a_i = 0$ or 1, and each edge joins sequences that differ in exactly one place.

5.3.2 Properties

For $Q_k = (V, E)$,

- $|V| = 2^k$
- $|E| = k2^{k-1}$
- It is regular of degree k.

6 Weighted Graphs

6.1 Definitions

- A weighted graph G is a graph in which each edge e is assigned a non-negative real number $w(e)$ called the weight of e.
- The weight of a subgraph H in G is the sum of the weights of the edges of H.

6.2 Dijkstra's Labelling Algorithm

Dijkstra's algorithm finds the length of the shortest path from a given vertex to every other vertex in a connected graph.

1. Let $A = v_1$. Assign A the permanent label $[0]$.
2. Until all vertices have been assigned permanent labels, do the following:
   (a) Suppose v_i is the vertex that has most recently acquired a permanent label, say $[l]$.
   (b) Examine each vertex v adjacent to v_i without a permanent label.
      i. If v is unlabelled, assign the temporary label $([l] + w(vv_i))$.
      ii. If v has a temporary label (t) and $[l]+w(vv_i) < t$, assign the temporary label $([l]+w(vv_i))$.
      iii. If v has a temporary label (t) and $t \leq [l] + w(vv_i)$, leave it unaltered.
   (c) Choose v_{i+1} to be a vertex with a temporary label that is the smallest of all current temporary labels. Change the label on v_{i+1} to a permanent one.

(See page 101 for an illustration)

6.3 Chinese Postman Problem

Given a connected, possibly weighted graph, the aim is to find the shortest closed walk that covers every edge at least once.

- If the graph is Eulerian, then the postman just follows an Eulerian trail.
- If the graph is not Eulerian, then certain edges will need to be used more than once.
   1. Identify the vertices with an odd degree.
   2. By the Hand-shaking Lemma, odd vertices occur in pairs, so list all partitions of pairs of vertices.
   3. Identify the partition with the least total weight, and add those edges to the graph.

7 Trees

7.1 Definitions

- A graph is *acyclic* if it has no cycles.
- A *tree* is a connected acyclic graph.
- An acyclic graph is a *forest* if the components of a forest are trees. A connected forest is a tree.

7.2 Properties of Trees

- Every edge of a tree T is a bridge. Conversely, if every edge of a connected graph is a bridge then it is a tree.
- If u and v are two non-adjacent vertices of a tree T, then $T + uv$ contains exactly one cycle C.
- A tree is a bipartite graph.
- In a tree, any two distinct vertices are connected by a unique path.
- The number of edges in a tree with n vertices is $n - 1$.

7.3 Theorems

- Every non-trivial tree has at least 2 end-vertices (vertices of degree 1).

7.4 Types of Trees

- *Path:* Can be straightened out into a line
- *Star:* $K_{1,m}$
- *Double star:* Contains exactly 2 vertices that are not end-vertices, which are adjacent
- *Caterpillar:* Removal of all end-vertices results in a path

7.5 Isomers of Paraffins

Paraffins are compounds with the formula C_kH_{2k+2}, where each hydrogen atom is bounded to exactly one carbon atom, and each carbon atom is bounded to four hydrogen atoms. To draw an isomer:

1. Draw a tree with k vertices labelled C, each with degree ≤ 4.
2. Add new edges with end-vertices labelled H at each C vertex until each C vertex has degree 4.

8 Labelled Trees

8.1 Definitions

- A *labelled tree* is one in which each vertex has a distinct label attached to it.

8.2 Isomorphism

Trees T_1 and T_2 whose vertices are labelled with the same set of labels are isomorphic if and only if for each pair of labels v and w, vertices v and w are adjacent in T_1 if and only if they are adjacent in T_2.

8.3 Cayley's Theorem

There are n^{n-2} distinct labelled trees on n vertices.

8.4 Prüfer Sequences

- A unique Prüfer can be constructed for each labelled tree:
    1. Find the end-vertex of $T_0 = T$ with the smallest label. Its neighbour is then the first term of the Prüfer sequence for T.
    2. Delete this end-vertex to produce T_1.
    3. Repeat the above steps until we arrive at $T_{n-2} = K_2$.

- The number of times a vertex v appears in the Prüfer sequence is $\deg(v) - 1$.
- A unique labelled tree can be constructed from a Prüfer sequence:
  1. A sequence of length $n-2$ produces a labelled tree with n vertices.
  2. Write out a sequence $S = 1, 2, \ldots, n$.
  3. Choose the smallest element in S that does not appear in the Prüfer sequence. Join the vertices with the corresponding labels and remove the used numbers.
  4. Repeat the procedure until 2 elements of the S remain. Join these vertices.

9 Spanning Trees

9.1 Definitions

- A *spanning subgraph* of a non-trivial graph G is a subgraph containing all the vertices of G.
- A *spanning tree* of G is a spanning subgraph of G that is a tree.

9.2 Properties of Spanning Trees

- Every connected simple graph G contains a spanning tree.
- Any spanning tree of a labelled K_n is a labelled tree on n vertices. By Cayley's Theorem, there are n^{n-2} distinct labelled trees on n vertices, so the number of distinct spanning trees on $K_n = n^{n-2}$.

10 Graphs and Matrices

10.1 Adjacency Matrix

The adjacency matrix $A(G) = (a_{ij})$ of a simple graph G is a $n \times n$ matrix with

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

A is

- binary (entries are 0 or 1)
- square
- symmetric
- $a_{ii} = 0$

10.2 Degree Matrix

The degree matrix $D(G) = (d_{ij})$ of a simple graph G is the $n \times n$ matrix with

$$d_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

10.3 Cost Matrix

The cost matrix $C = (c_{ij})$ of a simple weighted graph G is the $n \times n$ matrix with

$$c_{ij} = \begin{cases} w(v_i v_j) & \text{if } v_i,\ v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

10.4 Properties of Determinants of Matrices

- Transposition leaves the value of a determinant unaltered.
- If two rows or columns of a determinant are the same or proportional, the value of the determinant is 0.
- If two rows or columns of a determinant are interchanged, the value of the new determinant is the negative of the old.
- If the entries of a row or column of a determinant are divided by k, the new matrix is k times the old.
- A row or column operation leaves the value of a determinant unaltered.

10.5 Cofactors

If $A = (a_{ij})$ is an $n \times n$ matrix, the cofactor of the entry a_{ij} is $(-1)^{i+j}$ times the determinant of the $(n-1) \times (n-1)$ matrix found by removing the ith row and the jth column from A.

10.6 Matrix-Tree Theorem

Let G be a connected simple labelled graph with adjacency matrix A and degree matrix D. Then all cofactors of the matrix $M = D - A$ are equal, and their common value gives the number of spanning trees of G.

10.7 Counting Walks in a Graph

Let G be a simple graph or multigraph with adjacency matrix A. Then (a_{ij}) = number of edges $v_i v_j$ = number of walks of length 1 from v_i to v_j. Also, the (i, j)-entry of A^n is the number of walks of length n in G from v_i to v_j.

10.8 Counting Triangles in a Graph

The number of triangles $= \frac{1}{6} \times$ (sum of entries of the main diagonal of A^3) $= \frac{1}{6}$ trace(A^3).

11 Minimum Weight Spanning Trees

11.1 Kruskal's Algorithm

Given a connected, weighted graph G,

1. List the edges in order of increasing weight.
2. Make a copy of the graph without its edges.
3. Add an edge of least weight.
4. Continue to add edges of least weight such that cycles are not formed, until we have a spanning tree.

11.2 Prim's Algorithm

Given a connected, weighted graph G,

1. Make a copy of the graph without its edges.
2. Starting at a particular vertex, add an edge of least weight with a vertex that is in the partially constructed tree, and which does not result in the formation of a cycle.
3. Continue until a spanning tree is formed.

11.3 The Travelling Salesman Problem

A travelling salesman wishes to visit several cities and return to his starting point, covering the least possible total distance.

1. Suppose that G is a weighted Hamiltonian graph and H is a Hamiltonian cycle.

2. Remove any vertex v of G, leaving a subgraph G' of G, and a subgraph H' of H that is a spanning tree for G'.

3. Hence,

$$\begin{aligned} w(H) &= w(H') + \text{sum of weights of the two edges of } H \text{ incident with } v \\ &\geq w(\text{minimal spanning tree for} G') + \text{smallest sum of weights of 2 edges incident with } v \end{aligned}$$

12 Planar Graphs and Regions

12.1 Definitions

- A graph of G with n vertices and m edges is said to be *planar* if it can be drawn in the plane without crossings.
- Any such drawing is a *plane graph.*
- A *region* or face of G is a maximal portion of the plane for which any two points of the may be connected by a curve C such that each point of C is neither a vertex of G nor an edge of G.
- A simple planar graph G is called *maximal planar* if for every pair u, v of non-adjacent vertices of G, the graph $G + uv$ is non-planar.
- An *elementary subdivision* of a non-empty graph G is a graph obtained from G by removing some edge $e = uv$ and adding a new vertex w and edges uw and vw. This is equivalent to adding a vertex on an existing edge.
- A *subdivision* or a *homeomorph* of G is a graph H obtained from G by a succession of one or more subdivisions. H is said to be *homeomorphic* to G.
- An *elementary contraction* of a graph G is obtained by removing two adjacent vertices u and v and adding a new vertex w that is adjacent to those vertices to which u and v are adjacent.
- A graph G is *contractible* to a graph H if H can be obtained from G by a sequence of elementary contractions.

12.2 Properties

- Every subgraph of a planar graph is planar.
- Every graph with a non-planar subgraph must be non-planar.
- Every plane graph G contains an infinite (exterior) region.

12.3 Theorems

- *Euler's Formula:* If G is a connected plane graph with v vertices, e edges and f faces, then

$$v - e + f = 2$$

- If G is a plane graph with v vertices, e edges, f faces and k components, then

$$v - e + f = 1 + k$$

- If G is a simple maximal planar graph with $v \geq 3$ vertices and e edges, then
$$e = 3v - 6$$
- If G is a simple planar graph with $v \geq 3$ vertices and e edges, then
$$e \leq 3v - 6$$
- Every simple planar graph contains a vertex of degree ≤ 5.
- If G is a connected plane graph with v vertices, e edges and f faces in which each face is a n-cycle, then
$$e = \frac{n(v-2)}{n-2}$$
- Since the maximum number of edges in a plane graph occurs when each face is a triangle, we obtain a necessary condition for planarity in terms of the number of edges: If G is any simple planar graph with $v \geq 3$ vertices and e edges, then
$$e \leq 3v - 6$$
If G has no triangles, then
$$3 \leq 2v - 4$$
- The graphs K_5 and $K_{3,3}$ are non-planar.
- Any homeomorph of a graph G is planar or non-planar if and only if G is planar or non-planar.
- All cyclic graphs are homeomorphic.
- *Kuratowski's Theorem:* A graph G is planar if and only if G contains no subgraph isomorphic to K_5 or $K_{3,3}$ or a subgraph homeomorphic to K_5 or $K_{3,3}$.
- A graph is planar if and only if it contains no subgraph contractible to K_5 or $K_{3,3}$.

13 Polyhedra

13.1 Definitions

- A *polyhedron* is a surface whose faces are polygons.
- A polyhedron is *convex* if a straight line segment joining any two distinct vertices lies within the surface.
- For a convex polyhedron P, the number of vertices of degree k is V_k and the number of faces bounded by a k-cycle is F_k.
- A *regular convex polyhedron* is a polyhedron in which all the faces are congruent regular polygons and whose polyhedral angles are the same. That is, each face is bound by the same number of edges, and the degree of each vertex is the same.

13.2 Theorems

- *Euler Polyhedron Formula:*
$$V - E + F = 2$$
- Snce the degree of each vertex is at least 3, and each face is bounded by at least 3 edges,
$$2E = \sum_{k \geq 3} kV_k = \sum_{k \geq 3} kF_k$$
- Because every simple planar graph contains a vertex of degree ≤ 5, at least one face of every convex polyhedron is bounded by a k-cycle for some $k = 3, 4, 5$.
- There are exactly 5 regular convex polyhedra, called the *platonic solids*.

13.3 Duality

For a given graph G, the dual graph G^* is constructed as follows:

1. A vertex of G^* is placed in each region of G, including the exterior region.
2. Two distinct vertices of G^* are joined by an edge for each edge of G common to the boundaries of 2 corresponding regions of G.
3. A loop is added to a vertex of G^* for each bridge of G that belongs to the boundary of the corresponding region.
4. Each edge of G^* is drawn such that it crosses its associated edge of G but no other edge of G or G^*.

13.4 Properties of Dual Graphs

- If G is simple planar and connected, then so is G^*.
- If G is plane and disconnected, then G^* will be disconnected.
- $(G^*)^* \cong G$.
- G^* has a loop if and only if G has a bridge.
- If G is a plane graph, then G^* has multiple edges if and only if two faces of G have at least 2 edges in common.
- If G is a plane connected graph, then

$$\begin{aligned} v^* &= f \\ e^* &= e \\ f^* &= v \end{aligned}$$

14 Colouring Graphs

14.1 Definitions

- A graph G is *k-connected* if and only if the removal of fewer than k vertices results in neither a disconnected nor a trivial graph.
- A *map* is a 3-connected plane graph.
- A *proper colouring* of a graph is an assignment of colours to vertices such that no two adjacent vertices have the same colour.
- If a graph can be properly coloured with k colours, it is *k-colourable.*
- The *chromatic number* for a graph G, $\chi(G)$, is the minimum number of colours needed to properly colour G.
- A *contraction of an edge e* is performed by removing e and bring the two ends together. The resulting graph is denoted by $G\backslash e$.

14.2 Properties

- $\chi(K_n) = n$, $\chi(\bar{K_n}) = 1$, $\chi(K_{m,n}) = 2$, $\chi(K_n - v) = n - 1$, $\chi(C_{2n}) = 2$, $\chi(C_{2n+1}) = 3$
- If a graph contains an odd cycle, then $\chi(G) \geq 3$.
- Wheels with an even number of vertices are 4-chromatic.
- If G contains n vertices and has K_m $(m \leq n)$ as a subgraph, then $m \leq \chi(G) \leq n$.
- If $\chi(G) = k$, then G is m-colourable for all $m \geq k$.
- If G has v vertices, then $\chi(G) \leq v$.

14.3 Theorems

- If G is a simple graph with largest vertex-degree Δ, then G is $(\Delta + 1)$-colourable.
- *Brook's Theorem:* Let G be a simple connected graph with largest degree $\Delta \geq 3$. If G is not a complete graph, then G is Δ-colourable.
- Any simple planar graph is 4-colourable.

14.4 Welch-Powell Algorithm

The Welch-Powell Algorithm is used to assign colours to vertices of a graph

1. Order the vertices of G in decreasing degree.
2. Use one colour to paint the first vertex and, in sequential order, each vertex on the list not adjacent to a vertex previously painted with this colour.
3. Start again at the top of the list and repeat the process painting previously unpainted vertices using a second colour.
4. Continue with additional colours until all vertices have been painted.

14.5 Chromatic Polynomial

- The *chromatic polynomial*, $P_G(t)$, is the number of different colourings of a labelled simple graph G from t colours.
- $P_G(t) = 0$ if $t < \chi(G)$, and the smallest t such that $P_G(t) > 0$ is the chromatic number of G.
- $P_{K_n}(t) = t(t-1)(t-2)\ldots(t-n+1)$
- $P_{\bar{K}_n}(t) = t^n$
- If T is a tree on n vertices, then $P_T(t) = t(t-1)^{n-1}$
- $P_G(t) = P_{G-e}(t) - PG\backslash e(t)$ or $P_G(t) = P_{G+e}(t) + PG\backslash e(t)$
- The chromatic polynomial of a simple graph is indeed a polynomial.
- $P_G(t)$ has degree v.
- The coefficient of t^v in $P_G(t) = 1$.
- The coefficient of t^{v-1} in $P_G(t) = -e$.
- The constant term $= 0$.

- If G has k components, then $P_G(t) = \prod_{i=1}^{k} P_{G_i}(t)$, and the smallest power of t with a non-zero coefficient is k.
- Explicit form:

$$P_G(t) = \sum_{i=0}^{v-k} (-1)^i a_i t^{v-i}$$

15 Edge Colourings

15.1 Definitions

- An assignment of colours of a non-empty graph G such that adjacent edges are coloured differently is a *proper edge colouring* of G, or a *k-edge colouring* if k colours can be used.
- A graph is *k-edge colourable* if there exists an l-edge colouring of G for some $l \leq k$.
- The minimum k for which a graph is k-edge colourable is its *chromatic index*, $\chi'(G)$.
- A *class one graph* G is a non-empty graph with $\chi'(G) = \Delta(G)$, and a *class two graph* if $\chi'(G) = \Delta + 1$.

15.2 Properties

- If $\Delta(G)$ is the maximum vertex degree of a simple graph G, then

$$\chi'(G) \geq \Delta(G)$$

- *Vizing's Theorem:*

$$\Delta \leq \chi'(G) \leq \Delta + 1$$

- If $n \neq 1$ is odd, $\chi'(K_n) = n$, and if n is even, $\chi'(K_n) = n - 1$.
- If G is a simple bipartite graph with maximum vertex degree Δ, then $\chi'(G) = \Delta$.
- $\chi'(K_{m,n}) = \max\{m, n\}$.

15.3 Colouring K_n

If n is odd,

1. Draw K_n as a regular polygon.
2. The edges are paired off in a natural way, and colour all parallel edges the same colour.

If n is even,

1. Colour K_{n-1} as above.
2. Label the vertices v_i, $i \in \{1, 2, \ldots, n-1\}$ of K_{n-1} such that v_i is the vertex missing the colour i.
3. Add the extra vertex v and the edges $vv_i \in \{1, 2, \ldots, n-1\}$ with colour i.

16 Directed Graphs

16.1 Definitions

- A *directed graph* or *digraph* $D = (V, A)$ is a finite non-empty set V of objects called vertices together with a possibly empty collection of A of ordered pairs of elements of V.
- The elements of A are called *arcs* or *directed edges.*
- The graph obtained by changing arcs into edges is called the *underlying graph.*
- A digraph is called *simple* if all the arcs are different and there are no loops.
- Two vertices v, w of a digraph D are *adjacent* if either vw or wv is an arc.
- The *adjacency matrix* of the labelled digraph is the $n \times n$ matrix $A = (a_{ij})$, where a_{ij} is the number of arcs *from* v_i *to* v_j.
- The *out-degree* = $\text{outdeg}(v)$ of a vertex v is the number of arcs of the form vw.

$$\sum_{v \in V} \text{outdeg}(v) = \sum_{v \in V} \text{indeg}(v) = \text{number of arcs}$$

- A vertex with in-degree = 0 is called a *source* and a vertex with out-degree = 0 is called a *sink.*

16.2 Connectedness

- A digraph D is *connected* if the underlying graph of D is connected.
- A digraph D is *strongly connected* if for every pair u, v of vertices, D contains both uv and vu directed paths.
- A digraph D is strongly connected if and only if it is connected and every arc of D is part of a directed cycle.

16.3 Orientable Graphs

- A digraph D is called an *asymmetric digraph* or an *orientable graph* if whenever uv is an arc of D, then vu if not an arc of D.
- An *orientation* of G is obtained by adding arrows to the edges.
- A graph G is said to be *orientable* if there exists an orientation D that is strongly connected. D is then said to be a *strong orientation.*
- *Robbin's Theorem:* A graph is orientable if and only if it is connected and has no bridges.

16.4 Algorithm to Find a Strong Orientation

Given an orientable graph G,

1. Perform a depth-first search and create a DFS spanning tree, T.
2. For each edge e of G, orient e from smallest to largest vertex if e is in T, and from largest to smallest vertex if e is not in T.

17 Eulerian Digraphs

17.1 Definitions

- An *Eulerian trail* in a digraph D is an open trail of D containing all arcs and vertices of D.
- An *Eulerian circuit* in a digraph D is a circuit containing all arcs and vertices.
- A digraph that contains an Eulerian circuit is an *Eulerian digraph*
- A digraph that contains an Eulerian trail but not an Eulerian circuit is *semi-Eulerian.*

17.2 Theorems

- If D is an Eulerian digraph then it is strongly connected.
- A connected digraph D is Eulerian if and only if $\text{outdeg}(v) = \text{indeg}(v)$ for each vertex v of D.
- A connected digraph D is Eulerian if and only if its set of arcs can be split up into disjoint directed cycles.
- A connected digraph D is semi-Eulerian if there exist two vertices v and w such that:
  1. $\text{outdeg}(v) = \text{indeg}(v) + 1$
  2. $\text{outdeg}(w) = \text{indeg}(w) - 1$
  3. $\text{outdeg}(u) = \text{indeg}(u)$ for all vertices u, $u \neq v$, $u \neq w$

18 Hamiltonian Digraphs

18.1 Definitions

- A digraph D is *Hamiltonian* if it contains a spanning directed cycle, called a *Hamiltonian cycle*
- A non-Hamiltonian digraph D is *semi-Hamiltonian* if it contains an open path passing through every vertex, called a *semi-Hamiltonian path.*

18.2 Theorems

- If D is a strongly connected digraph with n vertices such that $\text{outdeg}(v) \geq \frac{n}{2}$ and $\text{indeg}(v) \geq \frac{n}{2}$ for all vertices v, then D is Hamiltonian.

19 Tournament

19.1 Definitions

- A *tournament* is a digraph D in which all pairs of vertices of D are joined by exactly one arc. That is, a tournament is obtained by orienting the edges of a complete graph.
- A tournament is *transitive* if whenever uv and vw are arcs of T, then uw is an arc of T.

19.2 Properties and Theorems

- A tournament of order n has $\binom{n}{2}$ arcs, and

$$\sum_{v \in V} \text{outdeg}(v) = \sum_{v \in V} \text{indeg}(v) = \binom{n}{2}$$

- A tournament is transitive if and only if it is acyclic.

- Every tournament has a Hamiltonian path.
- *Moser's Theorem:* Every strongly connected tournament with v vertices has a directed cycle of length $n = 3, 4, \ldots, v$.
- *Camion's Theorem:* A tournament is Hamiltonian if and only if it is strongly connected.
- A tournament T has a unique Hamiltonian path if and only if T is transitive.
- A non-decreasing seqauence of n non-negative integers is the score sequence of a transitive tournament if and only if the sequence is $0, 1, 2, \ldots, n-2, n-1$.
- For every positive integer n, there exists precisely one transitive tournament with n vertices.

19.3 Algorithm for Finding a Directed Hamiltonian Path

Build up the path vertex by vertex.