6 Graphs

6.1 Terminology and definitions

Reading: Rosen 9.1-9.4
Definition 1: A **graph** or an **undirected graph** $G = (V, E)$ consists of V, a set of **vertices** or **nodes**, and E, a set of **edges**, where an edge $e = (u, v)$ connects its **endpoins** u and v.
Definition 2: A **directed graph** $G = (V, A)$ consists of V, a set of vertices, and A, a set of **arcs**, where an arc $a = (u, v)$ goes from initial vertex u to terminal vertex v. Note that arc (u, v) and arc (v, u) are different.
Other types of graphs:

- Simple graph vs. multigraph
- Weighted graph vs. non-weighted graph
- Graph with self-loops

Examples of use of graphs: (1) Distance maps and (2) Precedence constraints.
More definitions and properties of graphs:

- The **degree** of a node, $deg(v)$, in an undirected graph is the number of edges incident with it.

  Handshaking Theorem: In an undirected graph $G = (V, E)$, $\sum_{v \in V} deg(v) = 2|E|$.

  Theorem: An undirected graph has an even number of nodes with odd degree. (Hint: $V = V_{even} \cup V_{odd}$.)

- In a directed graph, the **in-degree** of a node, $deg^-(v)$, is the number of arcs entering the node and the **out-degree** of a node, $deg^+(v)$, is the number of arcs leaving the node.

  Theorem: In a directed graph $G = (V, A)$, $\sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v) = |A|$.

Some special graphs:

- A **complete graph** is one in which there is an edge between every pair of vertices. For such a graph with n nodes, there are $\binom{n}{2} = \frac{1}{2}n(n-1)$ edges.
- A **bipartite graph** $G = (V, E)$, where V is partitioned into V_1 and V_2, and for any $e = (v_1, v_2)$, there must be $v_1 \in V_1$ and $v_2 \in V_2$.

  Theorem: G is bipartite if and only if G is 2-colorable, i.e., nodes in G can be colored with two colors such that no two nodes which are connected by an edge are given the same color.

Representing graphs:

- **Adjacency list:** In an undirected graph, for each vertex, create a list of all vertices connected to it by an edge. For example,

Vertex	Adjacent vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

  In a directed graph, for each vertex, create a list of all vertices reachable from it by an arc. For example,

Initial vertex	Terminal vertex
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

- **Adjacency matrix:** For a graph $G = (V,E)$ with $V = \{v_1, v_2, \ldots, v_n\}$, create an $n \times n$ matrix $A = [a_{ij}]$ such that $a_{ij} = 1$ if $(v_i, v_j) \in E$ and $a_{ij} = 0$ if $(v_i, v_j) \notin E$.

  For the undirected graph represented by the first adjacency list, using v_1, v_2, v_3, v_4, v_5 to replace the names of a, b, c, d, e respectively, its adjacency matrix is

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

  For the directed graph represented by the second adjacency list, its adjacency matrix is

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

- **Incidence matrix:** For an undirected graph $G = (V,E)$ with $V = \{v_1, \_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$, create an $n \times m$ matrix $A = [a_{ij}]$ such that $a_{ij} = 1$ if e_j is incident with v_i and $a_{ij} = 0$ if e_j is not incident with v_i.

  For the graph represented by the first adjacency list earlier, assuming v_1, v_2, v_3, v_4, v_5 are a, b, c, d, e respectively and $e_1 = (a,b), e_2 = (a,c), e_3 = (a,e), e_4 = (c,d), e_5 = (c,e), e_6 = (d,e)$, its incidence matrix is

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

Graph connectivity:
Definition: A **path** is a sequence of nodes, where there is an edge between any two consecutive nodes in the sequence. A **cycle** is a path that begins and ends at the same node. A path or cycle is **simple** if it does not contain the same node more than once.
Definition: An undirected graph is **connected** if there is a (simple) path between every pair of distinct nodes. A **connected component** of a graph is a connected subgraph that is not a proper subgraph of another connected subgraph.
Definition: A directed graph is **strongly connected** if there is a path from a to b and from b to a for any nodes a and b in the graph. A directed graph is **weakly connected** if the undirected graph obtained after removing directions on all arcs is connected. A **strongly connected component** of a directed graph is a strongly connected subgraph that is not contained in a larger strongly connected subgraph.

6.2 Euler and Hamilton paths/cycle

Reading: Rosen 9.5
Euler paths and cycles:

- An **Euler path** is a path that contains each edge exactly once. An **Euler cycle** is a cycle that contains each edge exactly once.

- The seven bridges of Konigsberg, a problem solved by Swiss mathematician Leonhard Euler.

- **Theorem:** A connected multigraph with at least two nodes has an Euler cycle if and only if each of it nodes has an even degree.

- **Theorem:** A connected multigraph has an Euler path but not an Euler cycle if and only if it has exactly two nodes of odd degree.

Hamilton paths and cycles:

- A **Hamilton path** is a path that passes throught each vertex exactly once. A **Hamilton cycle** is a cycle that passes through each vertex exactly once.
- A game called the "Icosian Puzzle", invented by Irish mathematician Sir William Rowan Hamilton, where there is a dodecahedron (a polyhedron with 20 vertices and 12 regular pentagons as faces) with each vertex representing a city and the objective of the puzzle is to travel along the edges to visit each city once to come back to the first city to make a tour.
- **Dirac's Theorem:** If G is a simple graph with $n \geq 3$ vertices such that $deg(v) \geq \frac{n}{2}$ for any vertex v in G, then G has a Hamilton cycle.
- **Ore's Theorem:** If G is a simple graph with $n \geq 3$ vertices such that $deg(u) + deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton cycle.
- The best algorithms **known** for finding a Hamilton path/cycle in a graph or determining no such path/cycle exists have exponential-time complexity, as these problems have been proved to be **NP-complete**, a class of problems for which no polynomial-time algorithms have been found.
- The **Traveling Salesman Problem** is a generalization of the Hamilton cycle problem, where the input is a weighted graph and the goal is to find a hamilton cycle with the minimum total weight.

6.3 A collection of graph problems

- Graph traversal: depth-first search and breadth-first search
- Topological sort in a directed acyclic graph:
- Shortest paths in a weighted graph: one-to-one, one-to-all, and all-to-all, Dijkstra's algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm
- Minimum spanning tree: Kriskal's algorithm and Prim's algorithm
- Network flow problem: Ford and Fulkerson algorithm
- Bipartite matching:
- Vertex cover, clique, dominating set, independent set:
- Graph coloring: