Chapter 7: Digraphs

22 Definitions

1. Definitions:

   - **Directed graph** or **digraph**
   - **Arcs** (directed edges)
   - **Underlying graph** (Undirected graph)
   - **Simple digraph:** arcs of D are all distinct, and no loops
     Note: The underlying graph of a simple digraph need not be a simple graph.
   - **Isomorphic:** isomorphism between underlying graphs-preserves arc-vertex ordering.
   - **adjacent**, **incident**
   - **walk from v_0 to v_m**; directed trail, path, cycle
   - **adjacency matrix:** $\mathbf{A} = (a_{ij})$, where $a_{ij} = \#$ of arcs from v_i to v_j
   - **connected:** D is connected if the underlying graph G is connected
   - D is **strongly connected** if there is a path from v to w for any vertices v, w
     Give an example of a connected digraph D that is not strongly connected.
   - Street map w/all one-way streets example: if connected, can get from any part of the city to any other (possibly illegally); if strongly connected, can get from any part of the city to any other legally.
   - G is **orientable** if each edge of G can be directed so that the resulting digraph is strongly connected.
     Note: G Eulerian $\Rightarrow$ follow any Eulerian trail $\Rightarrow$ G orientable.

2. **Theorem:** Let G be a connected graph. G is orientable $\Leftrightarrow$ each edge of G is contained in at least one cycle.
   pf. $\Rightarrow$

   - Pick an edge e, incident to v and w.
   - Assume WLOG that e goes from v to w.
   - Since there is a path from w to v, this creates a cycle with e.

   $\Leftarrow$ Assume each edge is contained in at least one cycle.

   - Choose a cycle C_1 and directly its edges cyclically.
   - If every edge of G is in C_1, we're done.
   - If not, pick an edge e not in C_1 but adjacent to an edge of C_1.
   - e is contained in some cycle C_2, and we cyclically direct edges in C_2 not already directed. (Illustrate)
   - The resulting digraph is strongly connected.
   - We continue by choosing an edge adjacent to a directed edge until all edges are directed.
   - Since the graph remains strongly connected at each stage, the result follows.

3. Critical Path Problem (weighted digraph-each arc represents length of time for activity):
   Go through with example.

   - Use shortest path algorithm, only modify to find longest path.
   - Longest path (**critical path**): any delay in an activity on this path creates a delay in entire job.
   - Can also calculate the latest time by which operation is completed if work is not delayed.

23 Eulerian Digraphs and Tournaments

1. Definitions:
   - **Eulerian digraph**, **Eulerian trail**
   - **out-degree**, **in-degree**
   - **handshaking dilemma:** sum of in-degrees must equal sum of out-degrees
   - **source:** in-degree 0; **sink:** out-degree 0
   - **Hamiltonian digraph**, **semi-Hamiltonian digraph**

2. **Theorem:** A connected digraph is Eulerian $\Leftrightarrow$ for each vertex v of D, $\text{outdeg}(v) = \text{indeg}(v)$.
   pf. Similar to proof for undirected graphs.

3. **Theorem:** Let D be a strongly connected digraph with n vertices. If $\text{outdeg}(v) \geq \frac{n}{2}$ and $\text{indeg}(v) \geq \frac{n}{2}$ for each vertex v, then D is Hamiltonian.

4. **Tournament:** a digraph in which any two vertices are joined by exactly one arc.
   Can be used to record the results of a round-robin tournament. (May have sources and sinks.)

5. **Theorem:**
   (i) Every non-Hamiltonian tournament is semi-Hamiltonian.
   (ii) Every strongly connected tournament is Hamiltonian.
   Proof in book.

6. Example: (p. 108, pr. 23.2)
   Give an example of a digraph for a tournament on 5 vertices.
   (a) Find cycles of length 3, 4, and 5.
   (b) Find an Eulerian trail.
   (c) Find a Hamiltonian cycle.

7. Example: (p. 108, pr. 23.3)
   Prove that a tournament cannot have more than one source or more than one sink.