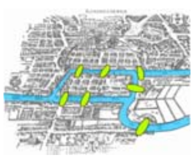
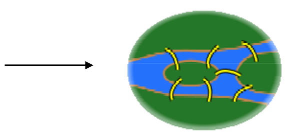
Section 5.1

Eulerian Graphs

KÖNIGSBERG BRIDGE PROBLEM





Images from https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

EULERIAN GRAPHS

- A graph (or multigraph) is **eulerian** if it has a circuit containing all the edges of the graph.
- The circuit itself is called an **eulerian circuit**.
- A trail containing every edge of a graph is called an **eulerian trail**.

EQUIVALENT STATEMENTS FOR EULERIAN GRAPHS

Theorem 5.1.1: The following statements are equivalent for a connected graph G.

1. The graph G contains an eulerian circuit.
2. Each vertex of G has even degree.
3. The edge set of G can be partitioned into cycles.

TWO COROLLARIES

Corollary 5.1.1: A connected graph G contains an eulerian trail if and only if at most two vertices of G have odd degree.

Corollary 5.1.2: Let $G = (V, E)$ be a connected graph with $2k$ $(k > 0)$ vertices of odd degree. Then E can be partitioned into exactly k open trails.

FLEURY'S ALGORITHM

Algorithm 5.1.1 Fleury's Algorithm.

Input: A connected (p, q) graph $G = (V, E)$.

Output: An eulerian circuit C of G.

Method: Expand a trail C_i while avoiding bridges in $G - C_i$, until no other choice remains.

FLEURY'S ALGORITHM (CONCLUDED)

1. Choose any $v_0 \in V$ and let $C_0 = v_0$ and $i \leftarrow 0$.
2. Suppose that the trail $C_i = v_0, e_1, v_1, \ldots, e_i, v_i$ has already been chosen:
   a. At v_i, choose any edge e_{i+1} that is not on C_i and that is not a bridge of the graph $G_i = G - E(C_i)$, unless there is no other choice.
   b. Define $C_{i+1} = C_i, e_{i+1}, v_{i+1}$.
   c. Let $i \leftarrow i + 1$.
3. If $i = |E|$
   then halt since $C = C_i$ is the desired circuit;
   else go to step 2.

FLEURY'S ALGORITHM AND EULERIAN CIRCUITS

Theorem 5.1.3: If G is eulerian, then any circuit produced by Fleury's Algorithm is eulerian.

HIERHOLZER'S ALGORITHM

Algorithm 5.1.2 Hierholzer's Algorithm.

Input: A connected graph $G = (V, E)$, each of whose vertices has even degree.

Output: An eulerian circuit C of G.

Method: Patching together of circuits.

HIERHOLZER'S ALGORITHM (CONCLUDED)

1. Choose $v \in V$. Produce a circuit C_0 beginning with v by traversing, at each step, any edge not yet included in the circuit. Set $i \leftarrow 0$.
2. If $E(C_i) = E(G)$;
   then halt since $C = C_i$ is the eulerian circuit;
   else choose a vertex v_i on C_i that is incident to an edge not on C_i. Now build a circuit C_i^* beginning at the vertex v_i in the graph $G - E(C_i)$. (Hence, C_i^* also contains v_i.)
3. Build a circuit C_{i+1} containing the edges of C_i and C_i^* by starting at v_{i-1}, traversing C_i until reaching v_i, then traversing C_i^* completely (hence, finishing at v_i) and the completing the traversal of C_i. Now set $i \leftarrow i + 1$ and go to step 2.