Euler Graph & Euler Tour

Joy Mukherjee

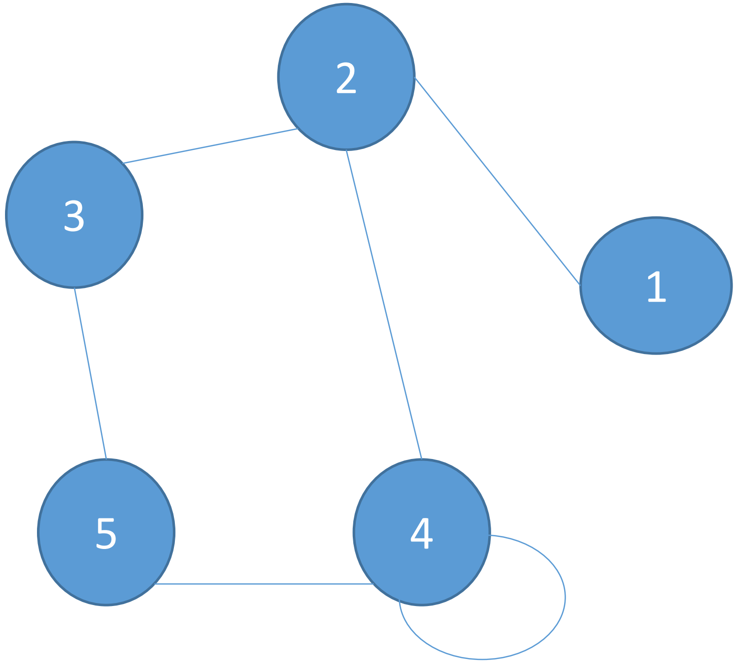
Walk and Trail

- A walk of length k is a sequence, v0,e1,v1,e2, …, ek, vk of vertices and edges such that $e_i = v_{i-1}v_i$ for all i
- A trail is a walk with no repeated edge
- A path is a walk with no repeated vertex
- A walk is closed if it has length at least one and its endpoints are equal
- A cycle is a closed trail in which “first = last” is the only vertex repetition
- A loop is a cycle of length one

Path & Cycle

- A path in a graph is a single vertex or an ordered list of distinct vertices v1, …, vk such that $v_{i-1}v_i$ is an edge for all $2 \leq i \leq k$.
- The ordered list is a cycle if $v_k v_1$ is also an edge.
- A path is an u, v-path if u and v are respectively the first and last vertices on the path.
- A path of n vertices is denoted by P_n, and a cycle of n vertices is denoted by C_n.
- A graph G is connected if it has a u, v-path for each pair u, v ∈ V(G).

Walk, Trail, Path, Cycle



Walk: 1 2 3 5 3 2 4 5

Closed Walk: 2 3 2

Trail: 1 2 3 5 4 4 2 (No repeated edge)

Closed Trail: 4 2 3 5 4 4

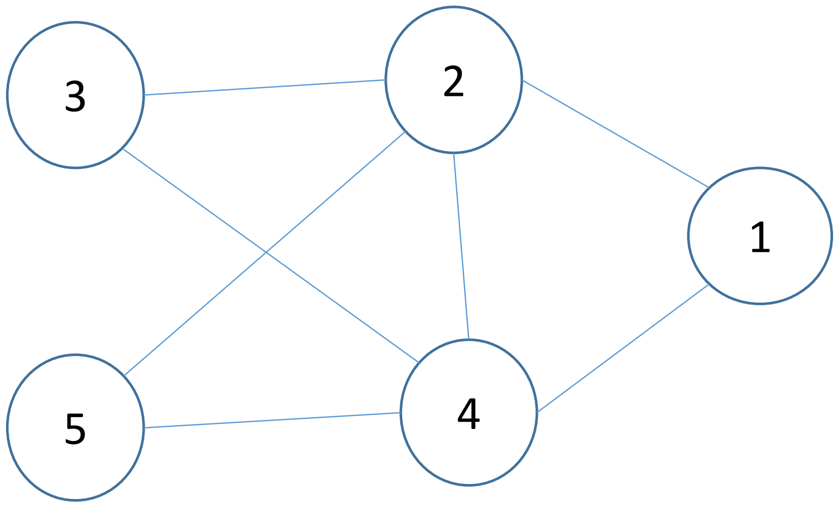
Path: 1 2 3 5 4 (No repeated vertex)

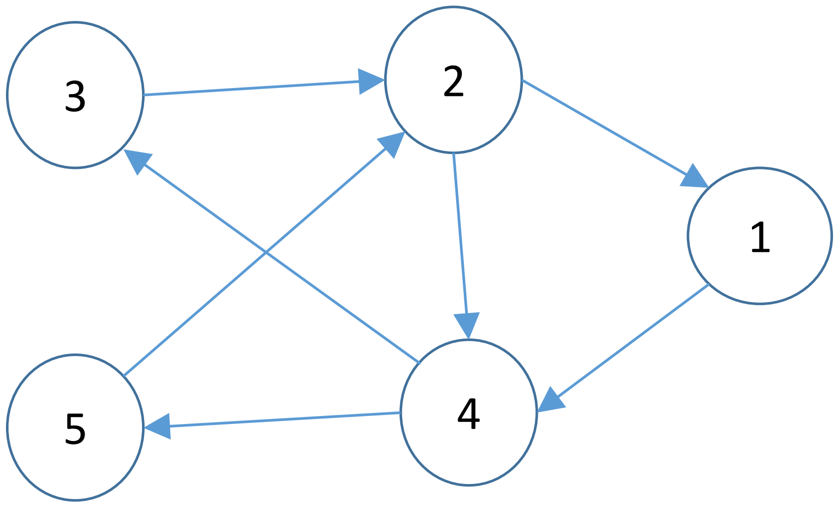
Cycle: 4 4 (self loop: cycle of length 1)

2 3 5 4 2 (Closed Path)

Euler Graph

- A trail is a walk with no repeated edges.
- A graph is Eulerian if it has a closed trail containing all edges.
- An even graph is a graph with vertex degrees all even.
- A vertex is odd (or even) when its degree is odd (or even).





1 4 5 2 4 3 2 1

Euler Trails in Directed Graphs

Input: A connected digraph G with $d^+(u) = d^-(u)$ for all $u \in V(G)$.

Output: A directed Euler trail

Algorithm:

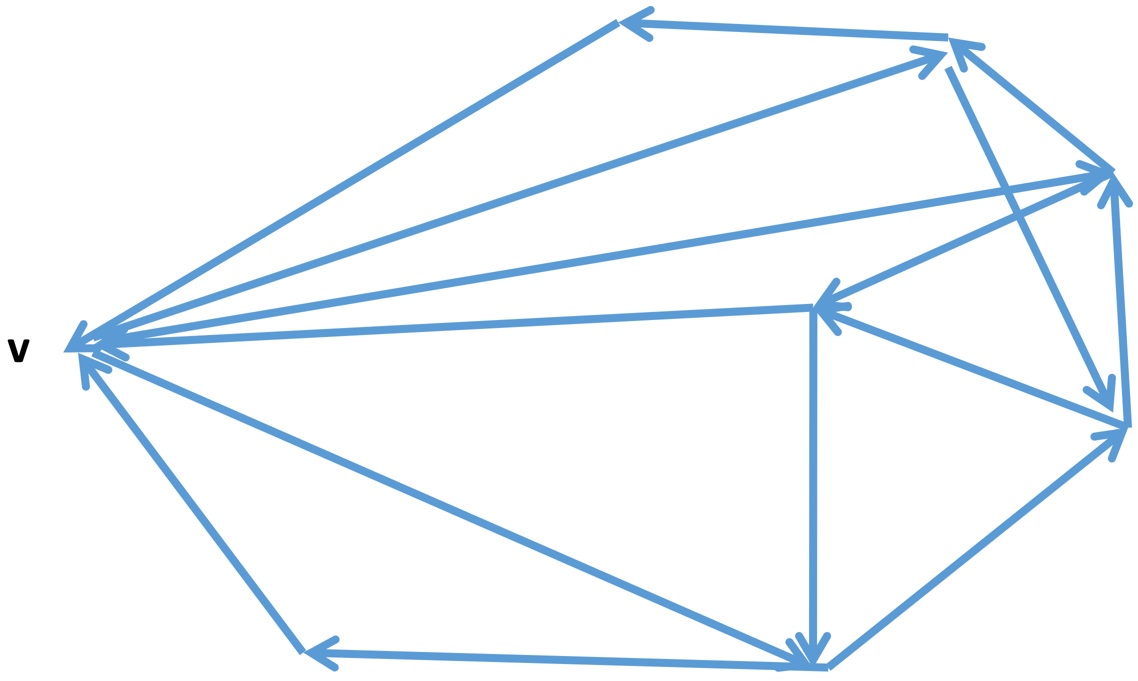
Step 1: G′ = Digraph obtained from G by reversing direction of each edge.

Step 2: DFS on G′ to construct T′ consisting of paths from s to V(G′)-s.

Step 3: Let T be the reversal of T′. T contains a u, s-path in G for each $u \in V(G)$.

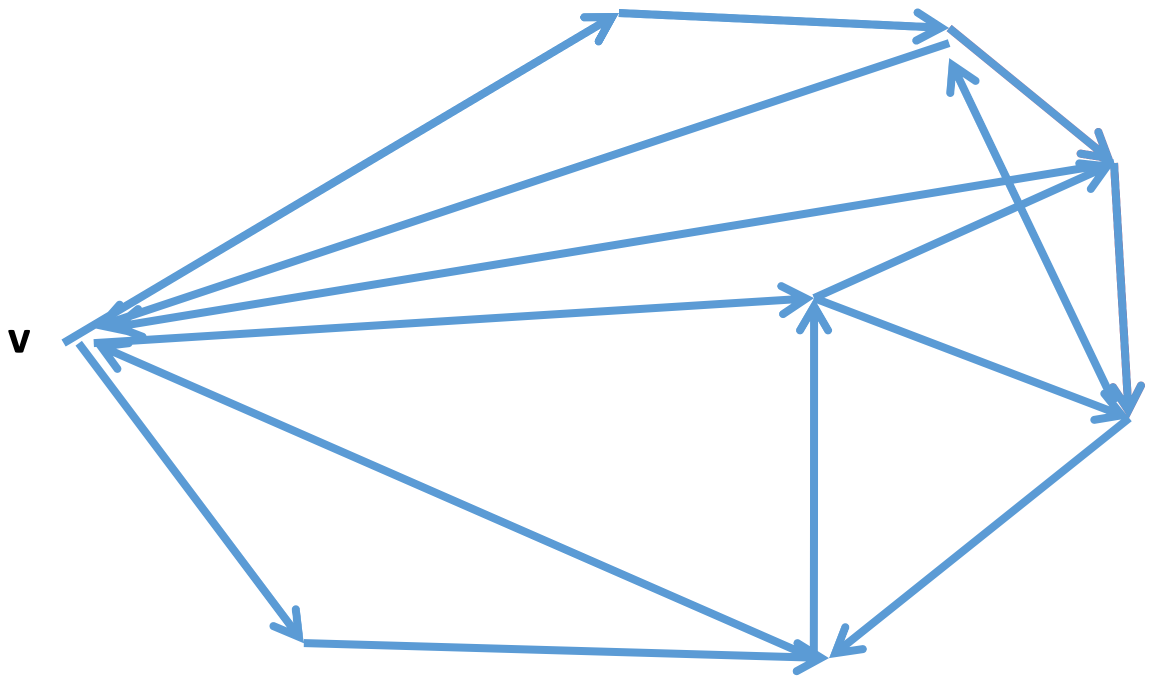
Step 4: Construct an Eulerian circuit from s as follows: Whenever u is the current vertex, exit along the next unused edge of G - T as long as possible. If there is no other edge from u, then use edges in T.

Euler Trails in Directed Graphs



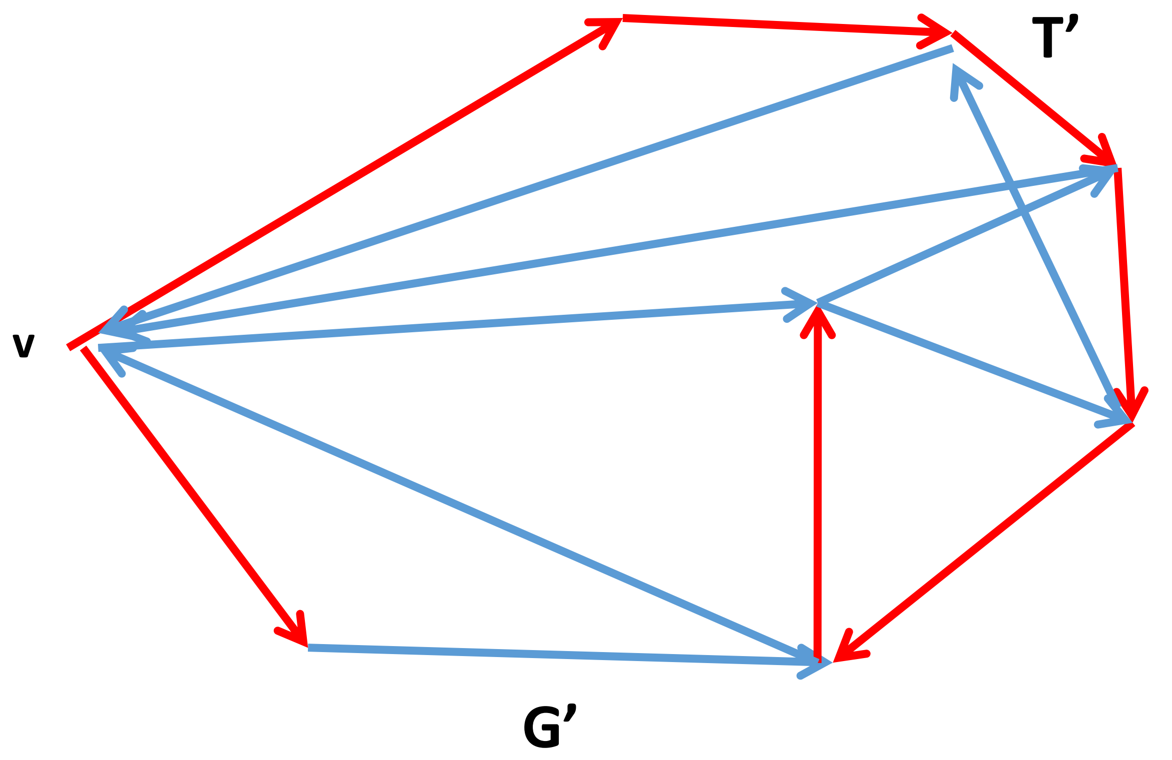
G

Reverse the edges



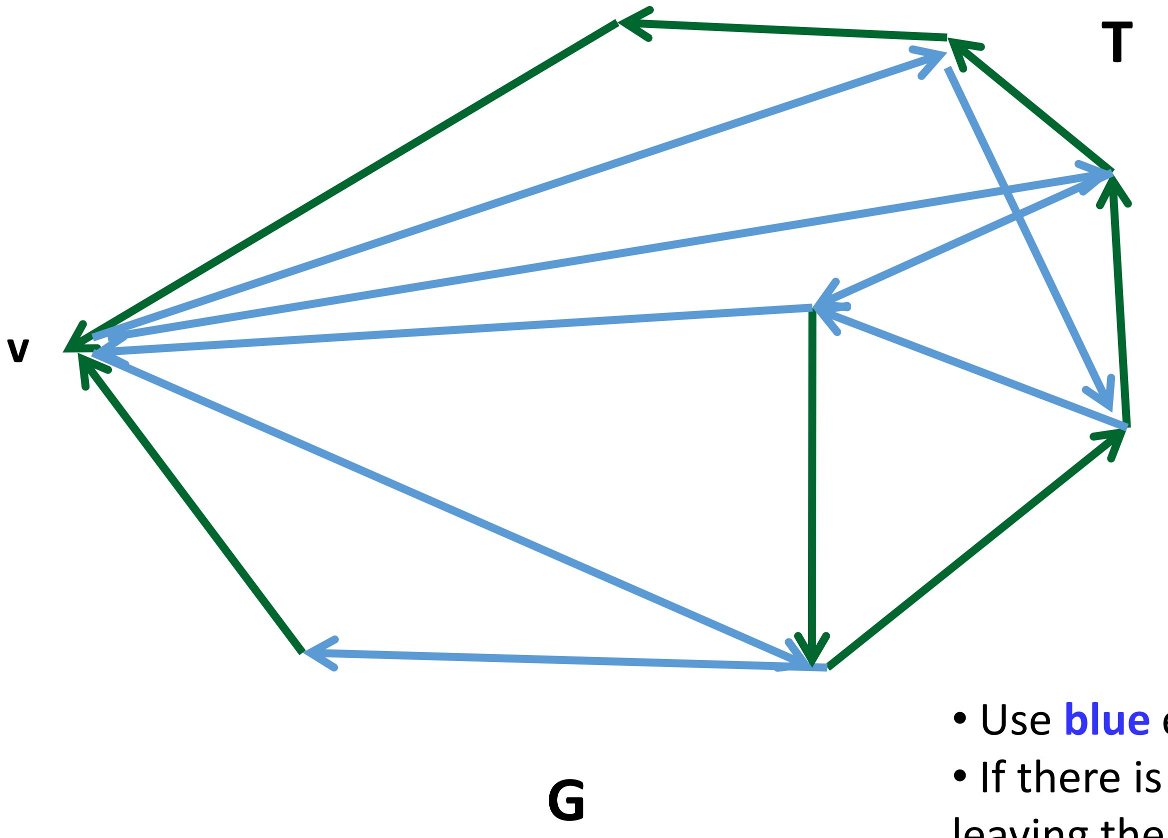
G’

DFS



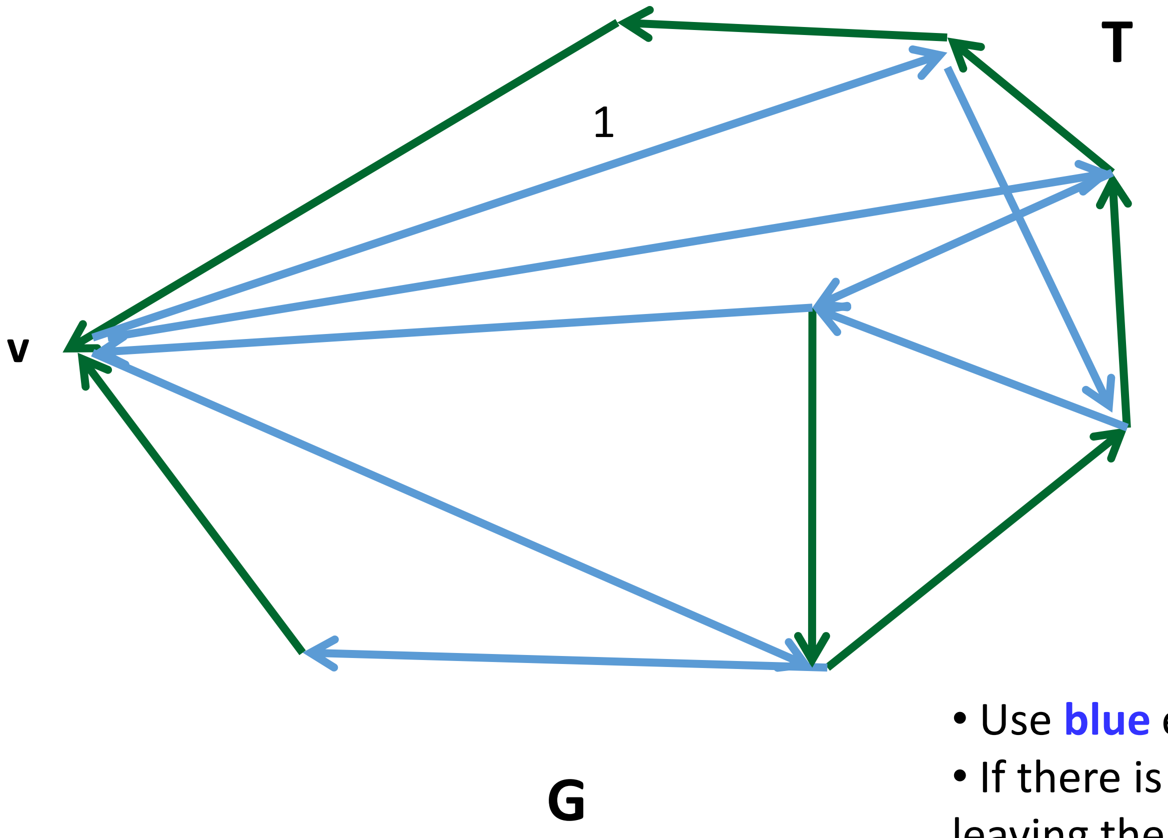
- DFS Tree T’ consists of **red** edges

Eulerian Circuit Construction



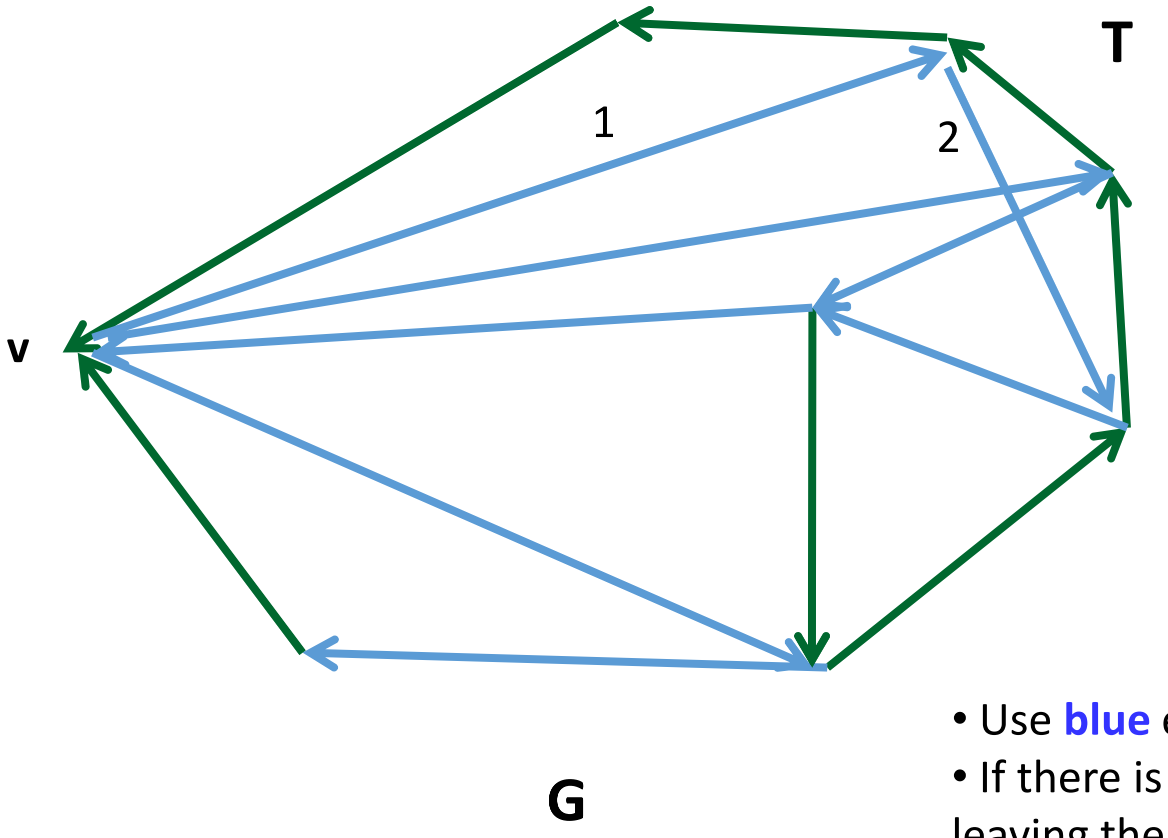
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



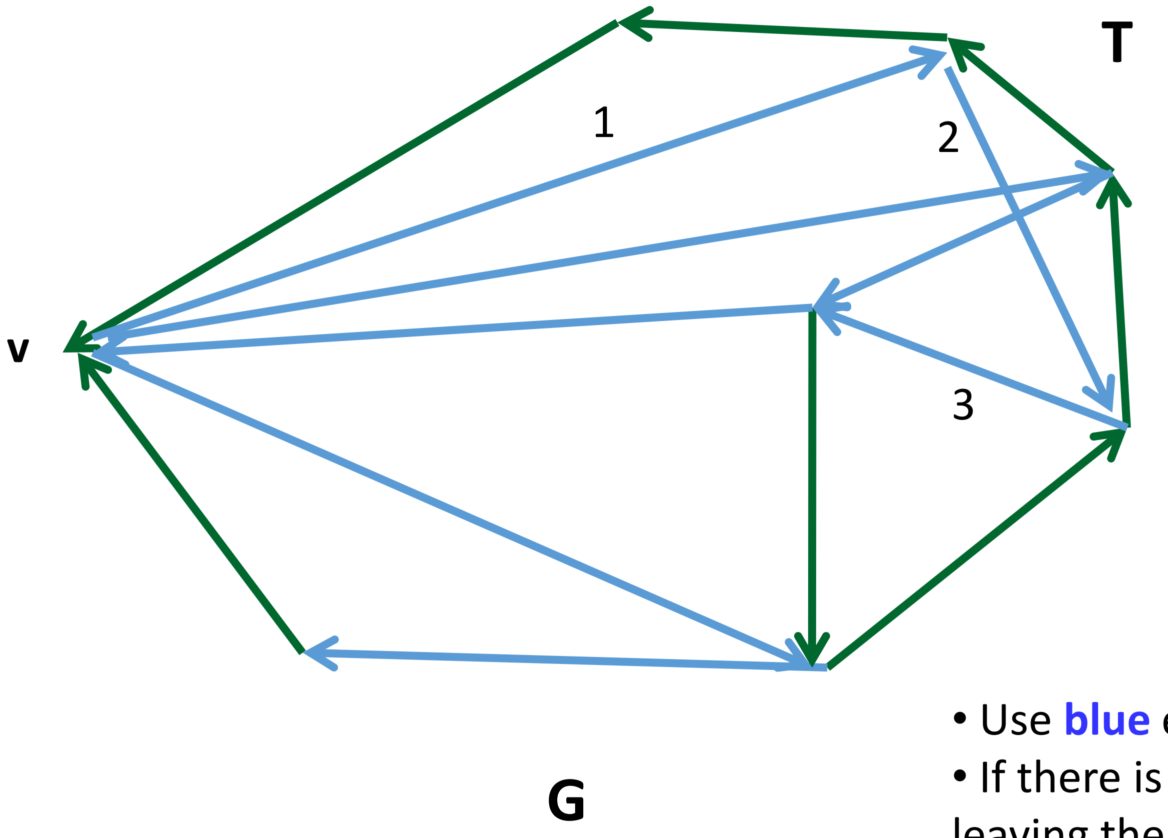
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



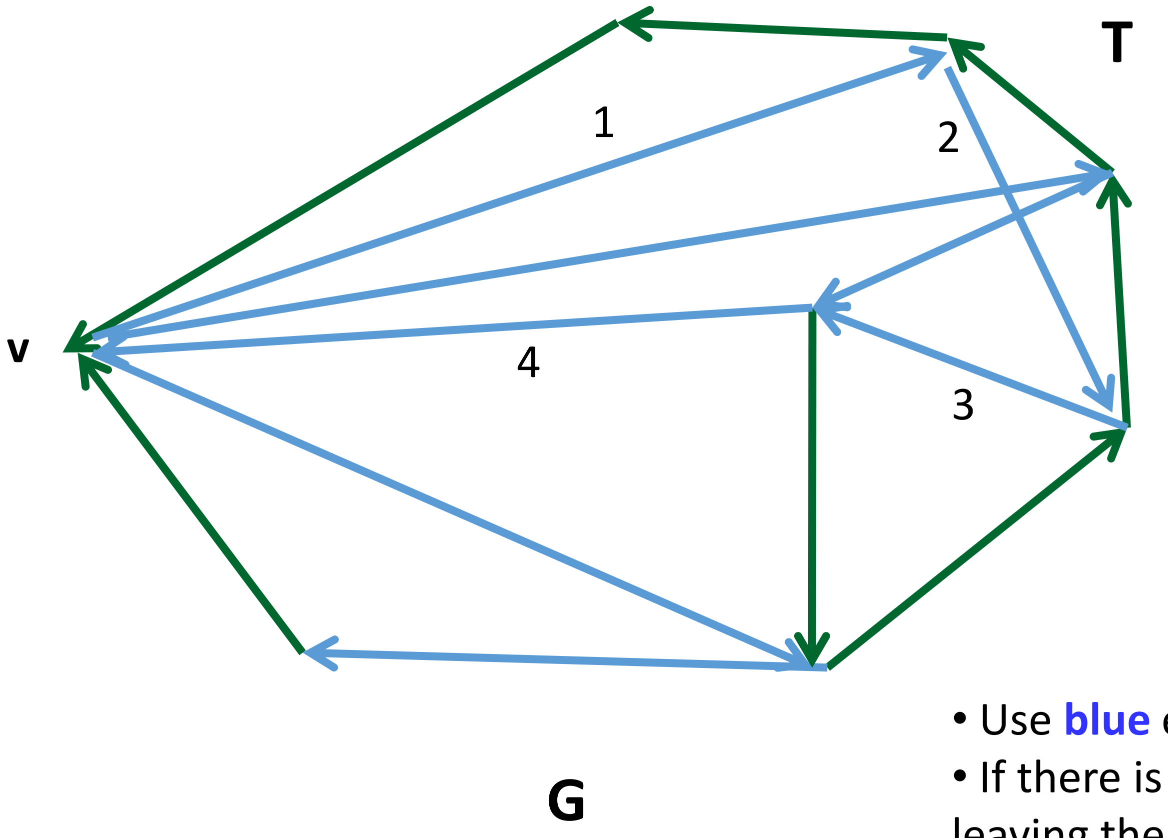
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



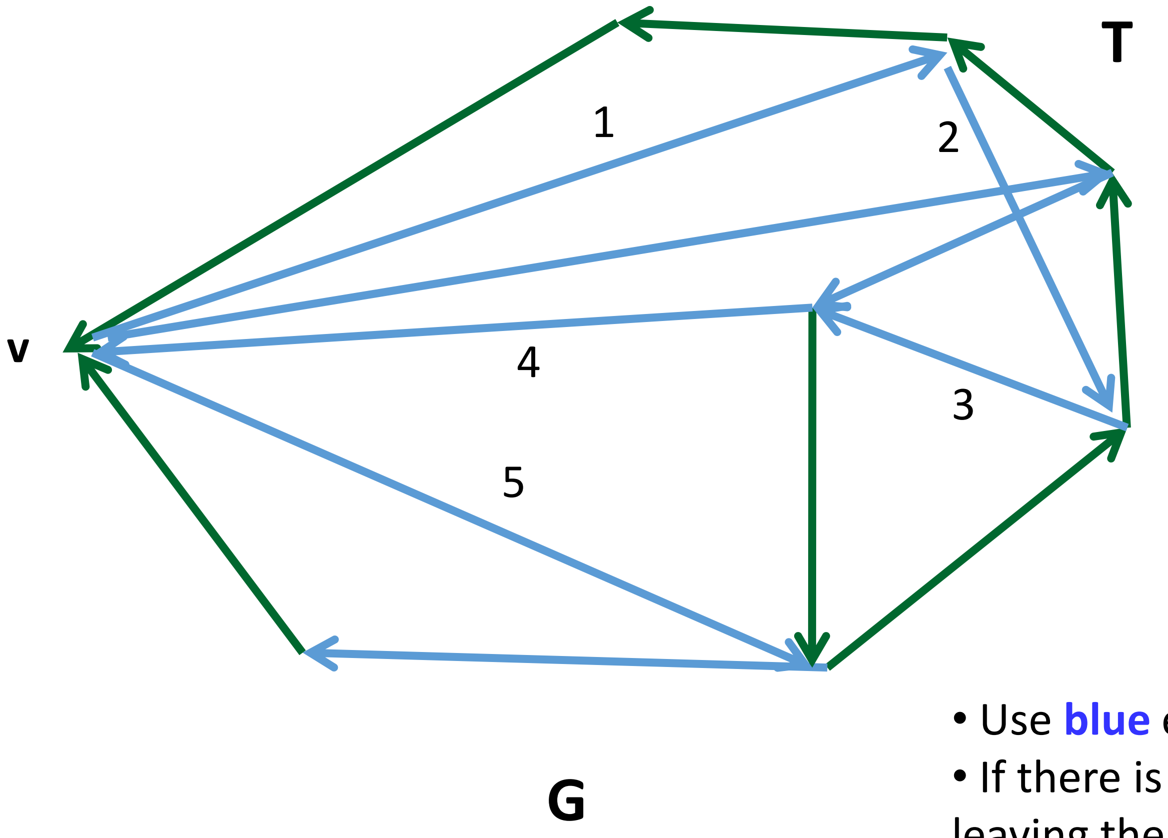
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



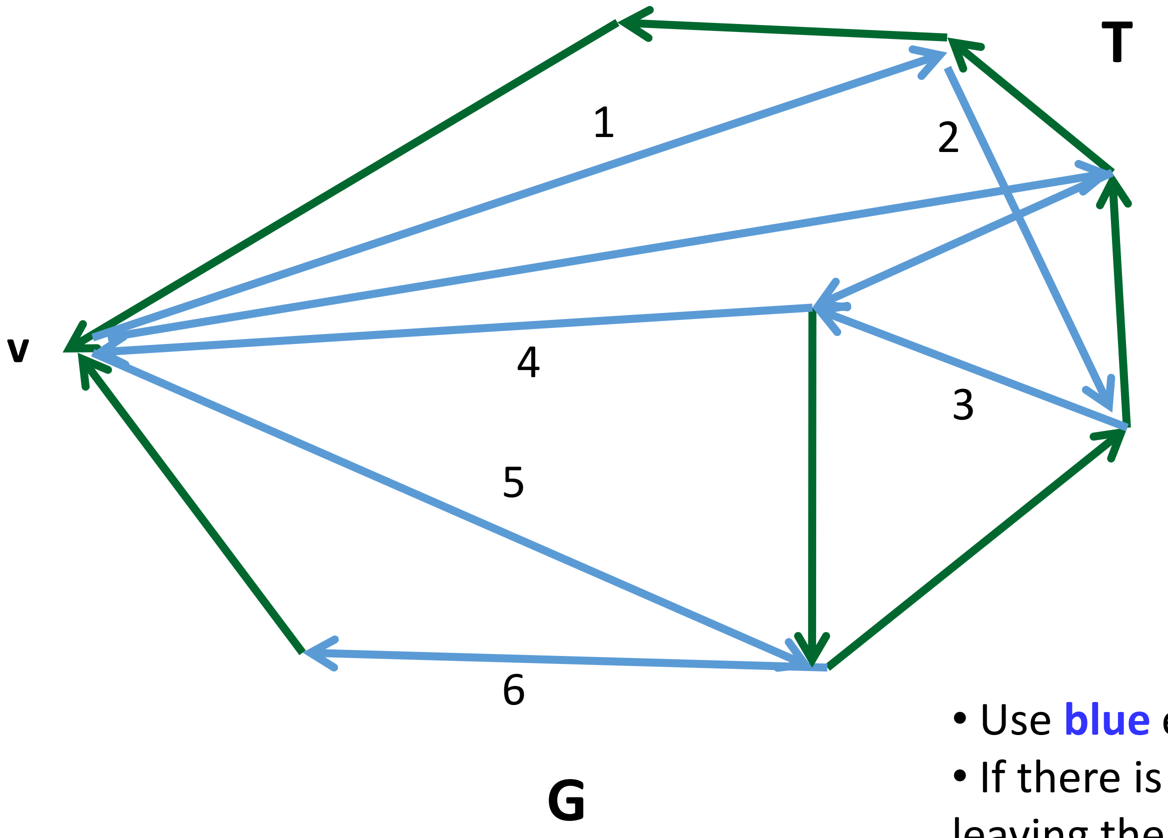
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



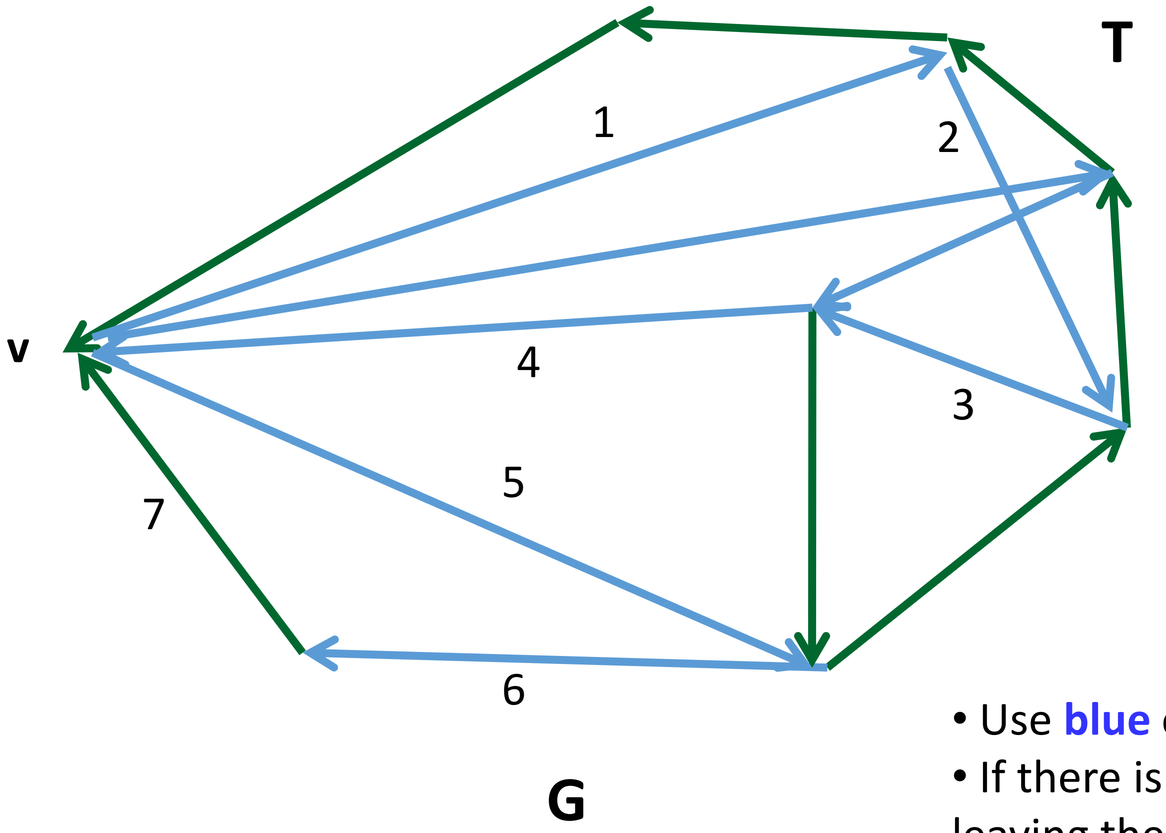
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



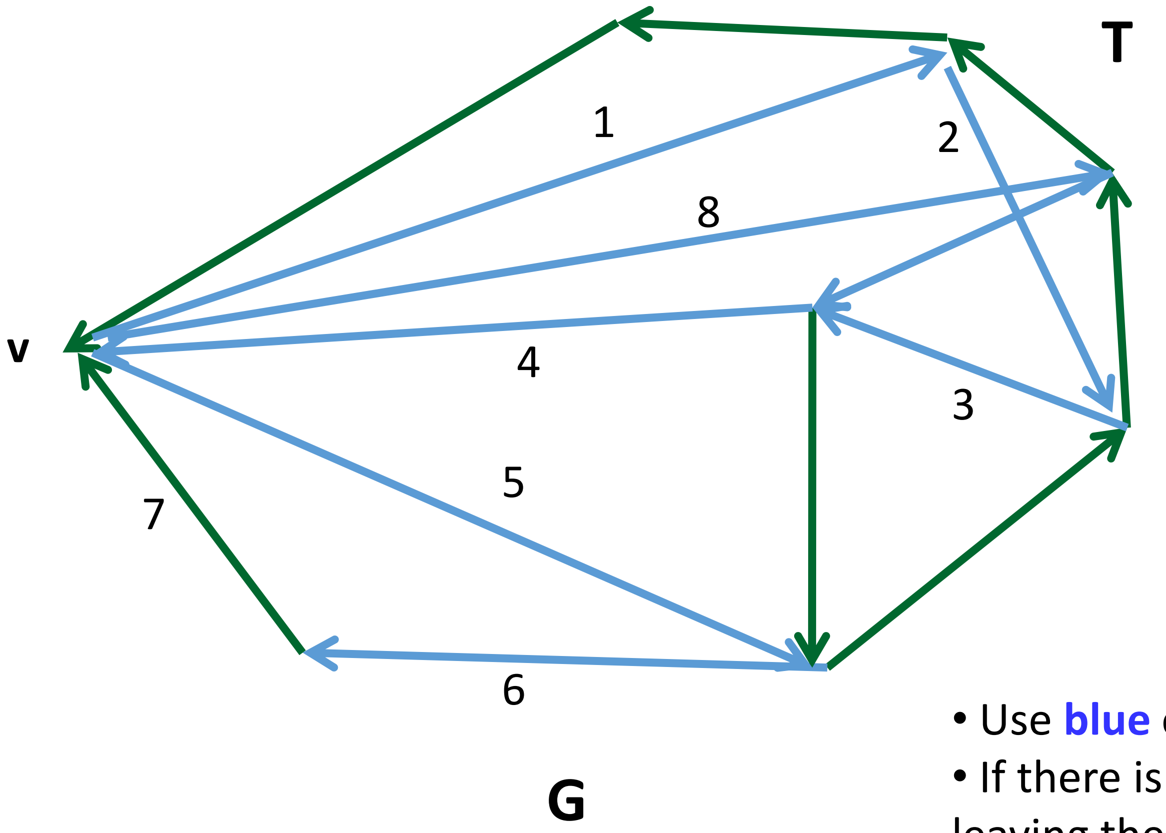
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



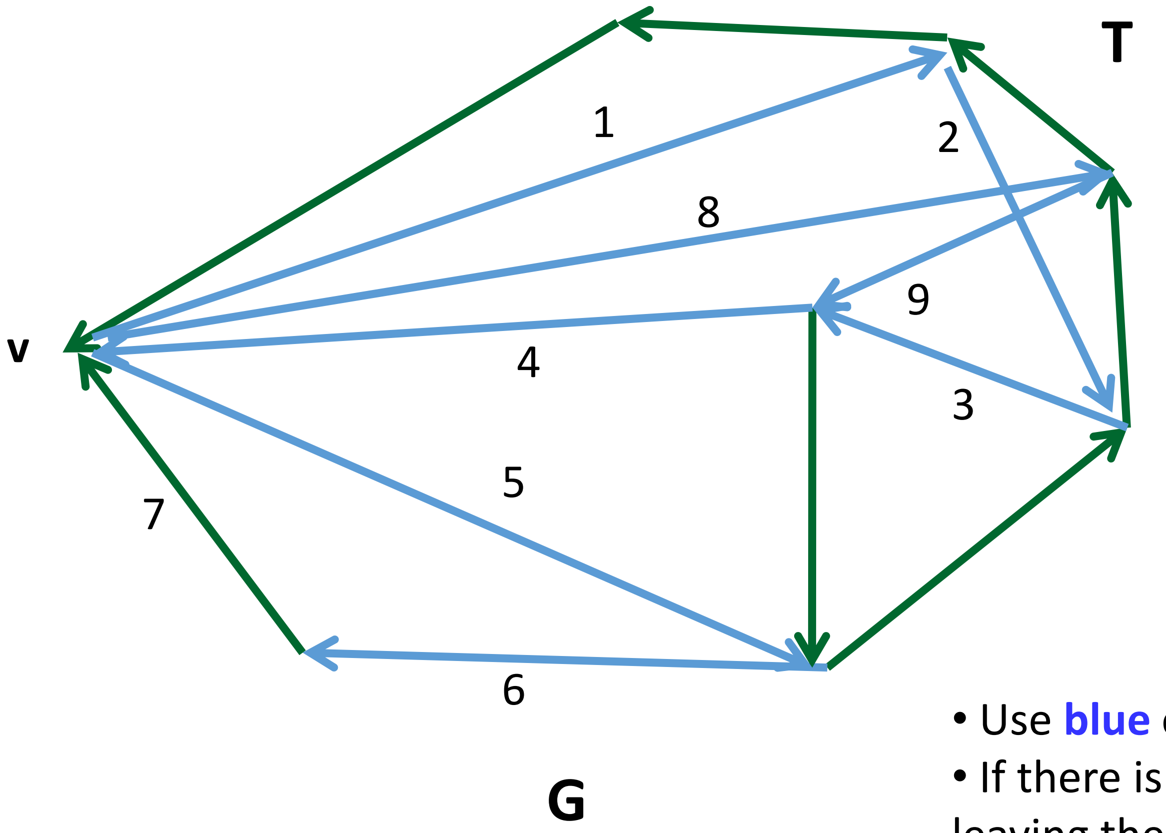
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



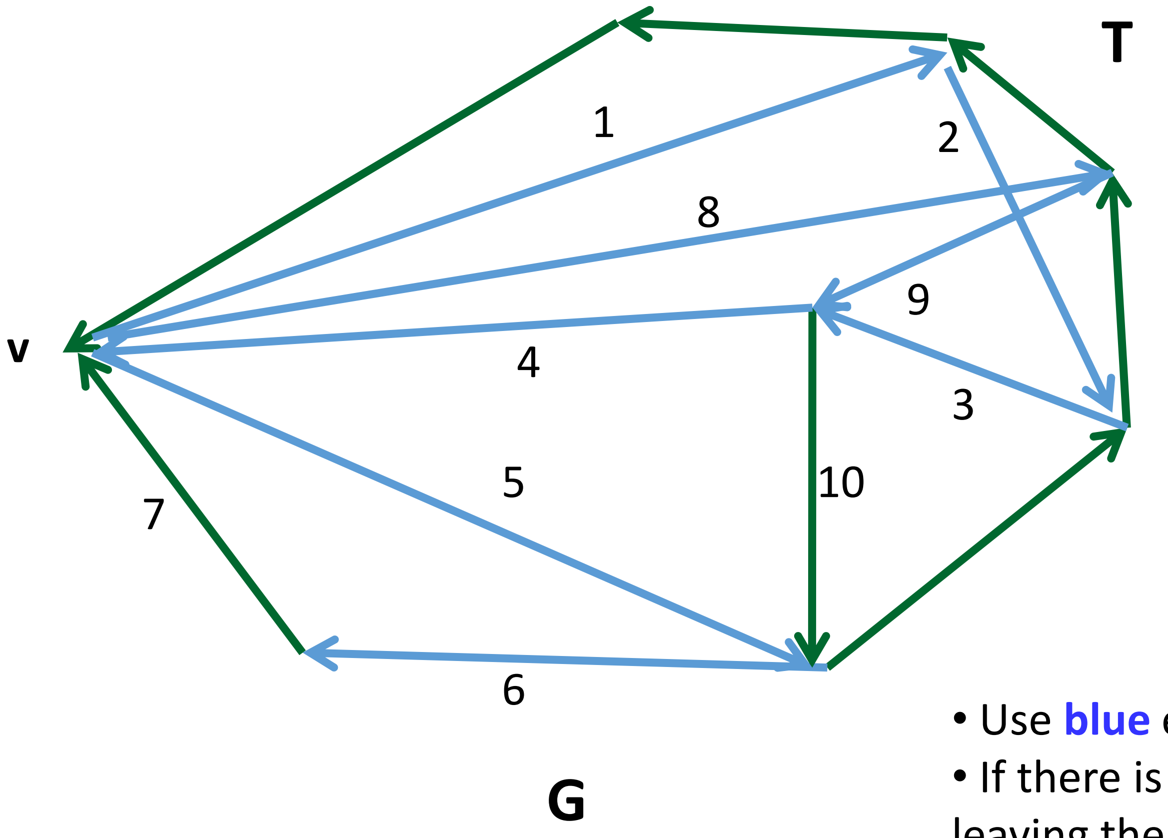
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



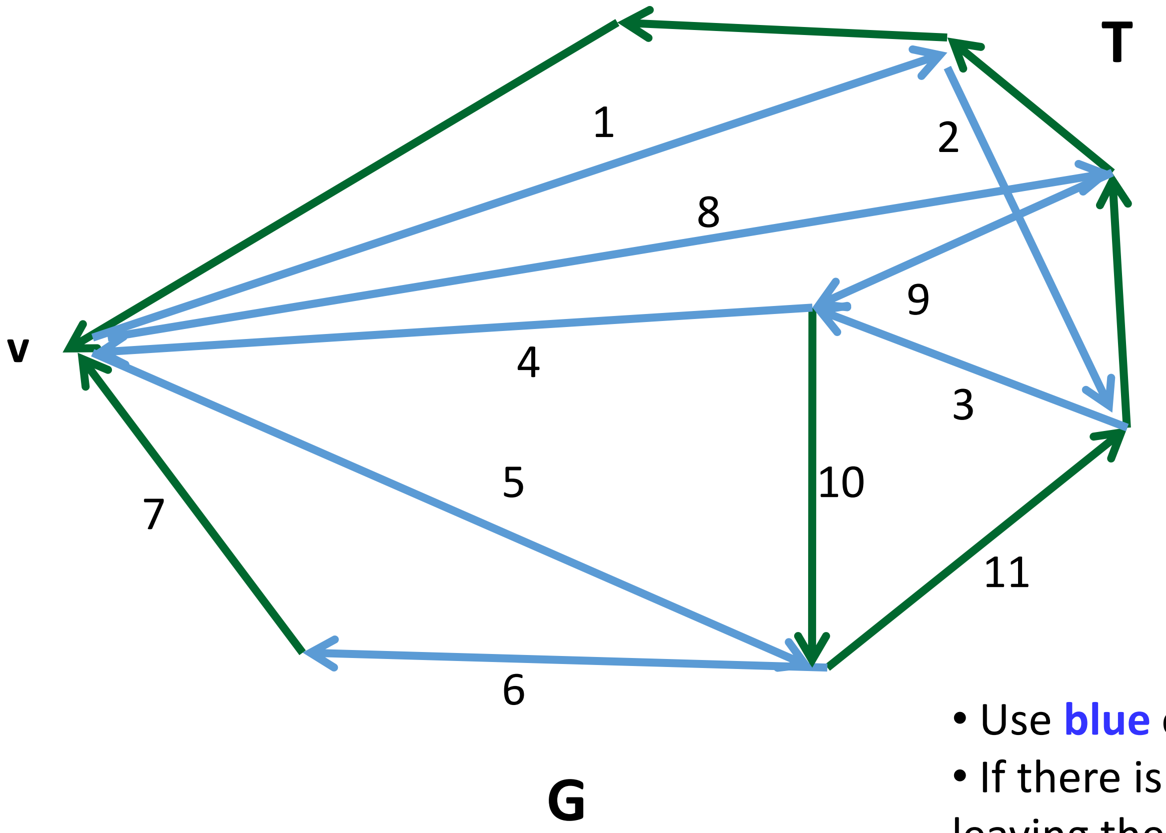
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



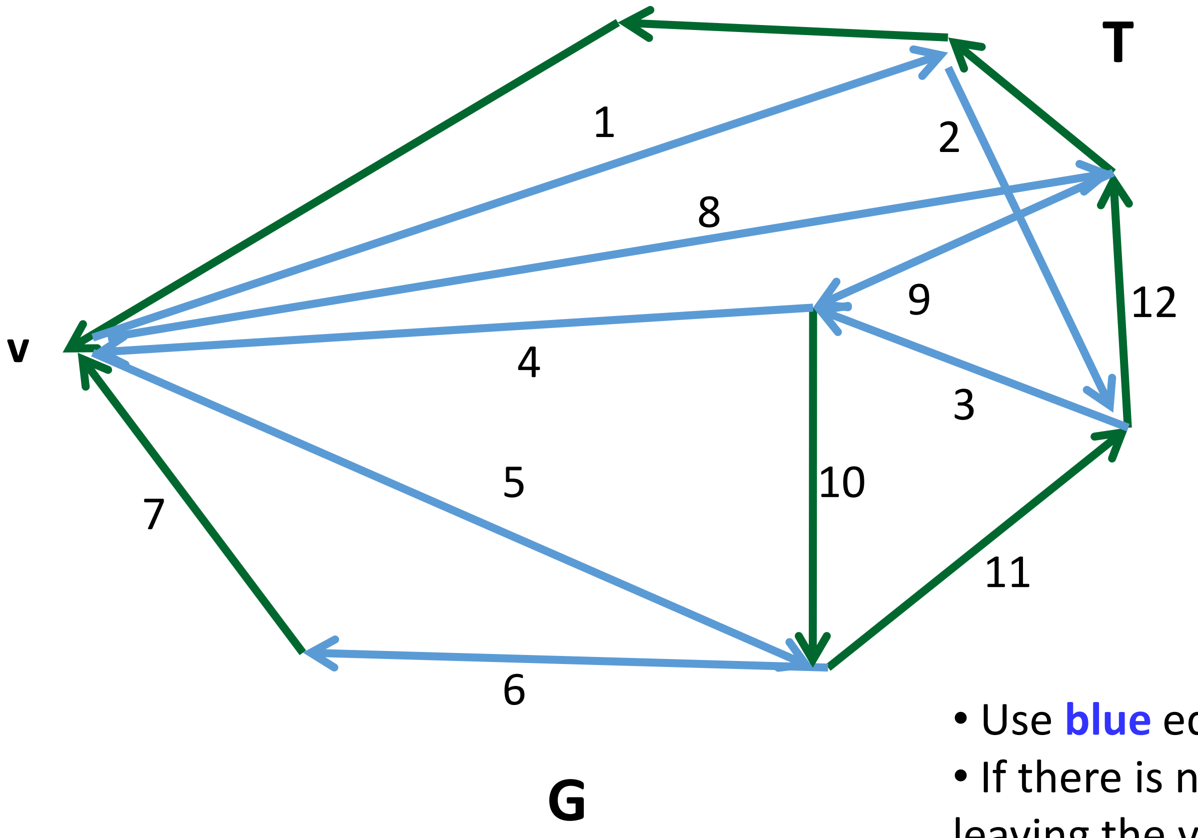
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



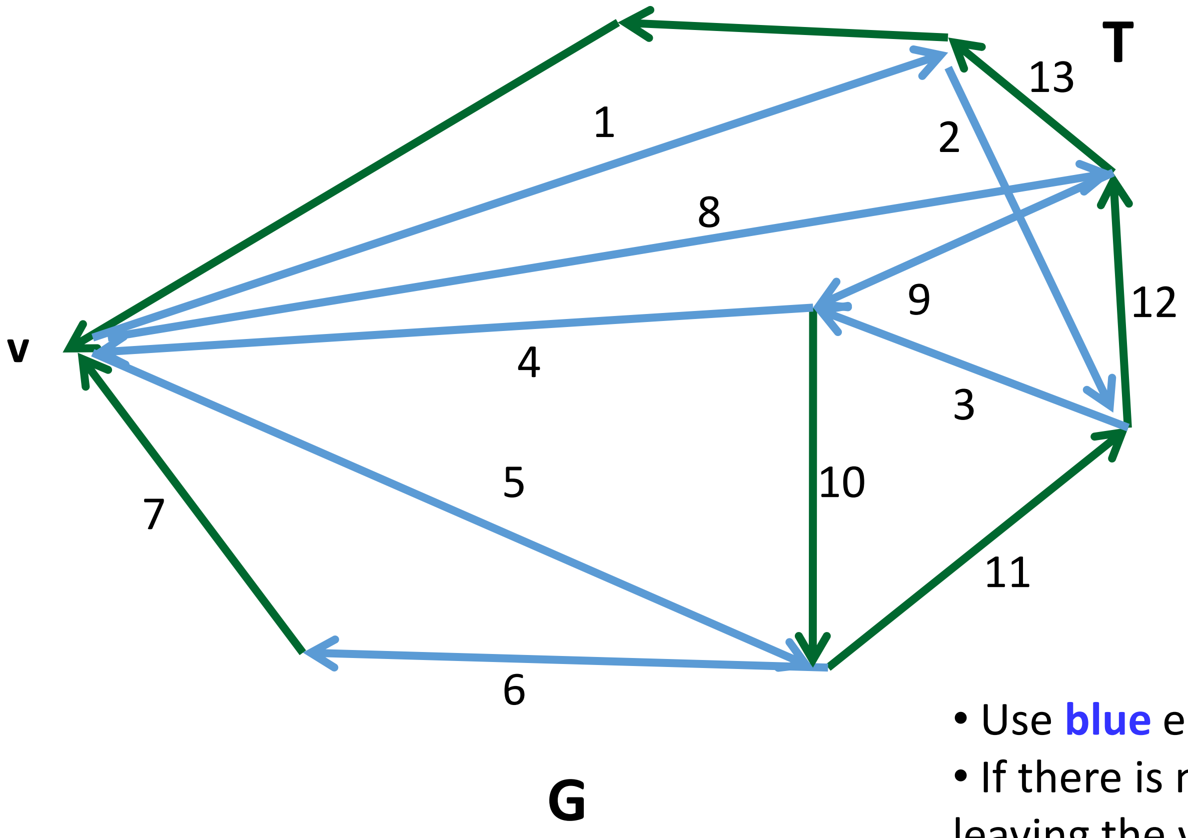
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



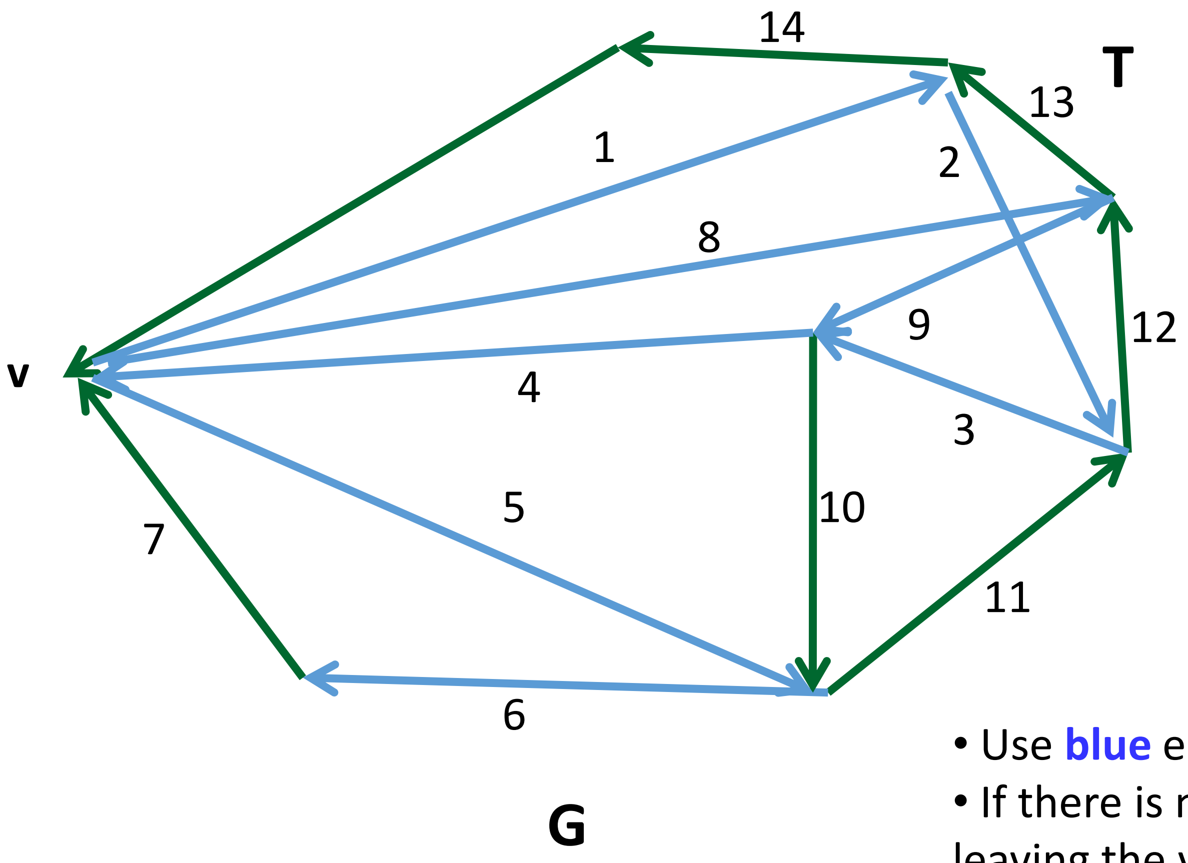
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



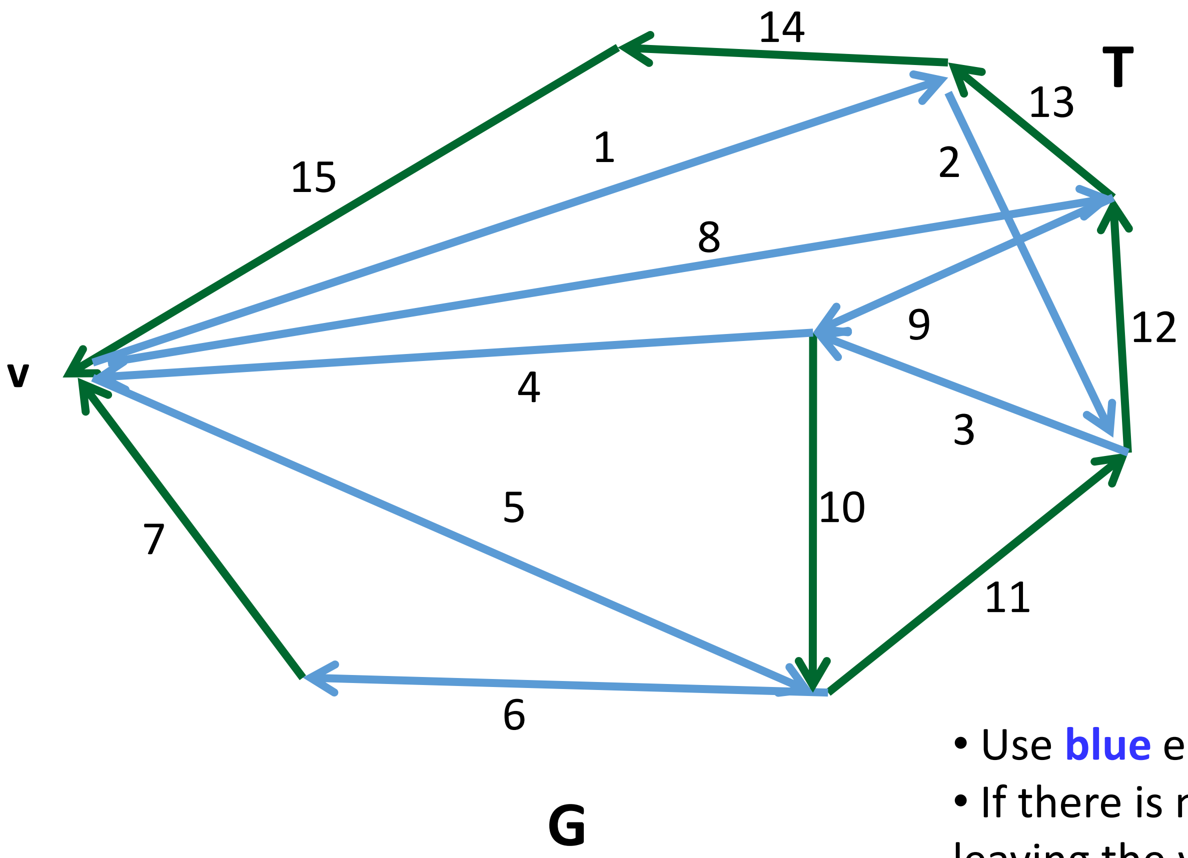
- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges

Eulerian Circuit Construction



- Use **blue** edges for leaving a vertex.
- If there is no **blue** edges left for leaving the vertex, use **green** edges