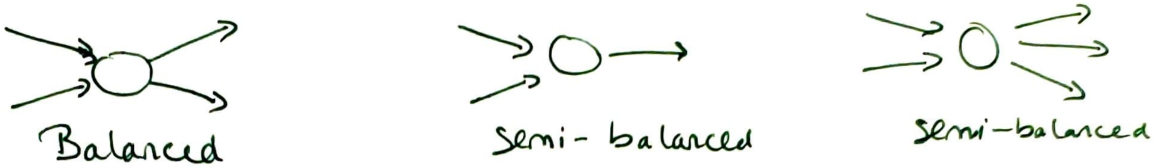
CS 342: Bioinformatics

Eulerian Cycles and Paths — Spring 2020

An **Eulerian path (cycle)** in a graph G is a path (cycle) that visits every edge in G exactly once.

A vertex is **balanced** if it has the same number of incoming and outgoing edges.

A vertex is **semi-balanced** if indegree differs by outdegree by 1.



Theorem: A connected directed graph G contains an Eulerian cycle if and only if all the vertices are balanced.

Proof. This is a just a proof-sketch!!!

($\Rightarrow$) Suppose G contains an Eulerian cycle. We need to show that all vertices in G are balanced.

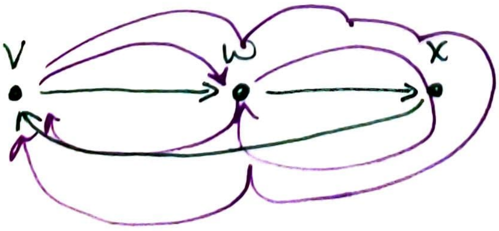
For any vertex v and any edge entering v, there is a corresponding edge leaving v. In particular, these are the edges used in the Eulerian cycle. Thus $in(v) = out(v)$.

($\Leftarrow$) Suppose all vertices in G and balanced. We need to show that G contains a Eulerian cycle.

We will do this by showing how to construct such a cycle.

- **Step 1:** Start at some vertex v. Keep traveling on unused edges from v until you hit a dead end. We claim that the dead end must be at v since the graph is balanced. This produces a cycle.
- **Step 2:** If the cycle we just found is not Eulerian, then it must contain some vertex w with untraversed edges (because the graph is connected). Repeat step 1 from w.
- **Step 3:** Combine all cycles found into one Eulerian cycle.

$\square$



Theorem: A connected, directed graph G has an Eulerian path if and only it contains at most two semi-balanced vertices and all other vertices are balanced.

- The path must start and end at the semi-balanced vertices.
- One vertex v has $in(v) - out(v) = 1$ and one vertex w must have $out(w) - in(w) = 1$
- Add edge from v to w and search for Eulerian cycle. Remove the edge (v, w) to get Eulerian path.