EULERIAN CYCLES AND PATHS

MATH CIRCLE (HS1) 2/1/2015

Warmup/Terminology

Recall that a graph $G = (V, E)$ is made up of vertices (V) and edges (E). This quarter we've dealt with *directed* graphs, i.e. graphs with edges that have a direction. Today we'll also look at *undirected* graphs, where the edges do not have directions.

Recall we can think of graphs as cities (the vertices) connected by roads (the edges). In this case, a directed graph has only one-way roads, while an undirected graph has only two-way roads.

In an undirected graph, the number of edges containing a vertex v is called the degree of v, denoted $d(v)$.

In a directed graph, the number of edges going out of a vertex v is called the out-degree of v, denoted $od(v)$. Similarly, the number of edges going into a vertex v is called the in-degree of v, denoted $id(v)$.

1) Prove the following, where $G = (V, E)$ is a graph.

a) If G is an undirected graph, $\sum_{v \in V} d(v) =$ twice the number of edges in G.

b) If G is a directed graph, $\sum_{v \in V} id(v) = \sum_{v \in V} od(v) =$ the number of edges in G.

Recall that an *Eulerian path* in a graph G is a path through ALL of the edges in the graph (using each edge only once). If this path starts and ends at the same point, it is called an *Eulerian cycle.*

We say a vertex u is *connected* to a vertex v if there is a path from u to v in G. Note: Order is important here! In a directed graph, it is possible for u to be connected to v but v is not connected to u.

An undirected graph is *connected* if any two vertices of G are connected.

Similarly, a directed graph is *strongly-connected* if any two vertices of G are connected.

A directed graph is *weakly-connected* if the underlying undirected graph (i.e. the same graph with the direction of edges removed) is connected.

2) a) If an undirected graph G has an Eulerian cycle, is it connected? What about the converse (i.e. if it is connected, does it have an Eulerian path)?

b) If a directed graph G has an Eulerian cycle, is it strongly-connected? What about the converse? Repeat with weakly-connected instead of strongly-connected.

c) Repeat a)+b) with Eulerian paths instead of Eulerian cycles.

Undirected Graph Case

1) For each of the undirected graphs on the graph examples page:

a) Label the degree of each vertex.

b) State whether the graph has an Eulerian cycle, an Eulerian path, or neither.

2) Using 1), try to come up with a rule as to when a connected graph has a Eulerian cycle, an Eulerian path, or neither. That is, fill in the following blanks:

a) A connected undirected graph G has an Eulerian cycle if and only if ____________.

b) A connected undirected graph G has an Eulerian path if and only if ____________.

3) a) Prove the forward ($\Rightarrow$) direction for both a) and b) of 2.

b) Assume you have proven both directions of 2a) (you'll prove this later). Prove 2b).

4) Algorithm for finding Eulerian cycles:

(1) Choose a starting vertex v.
(2) Follow (arbitrary edges) from v, until you get back to v. If all edges are used, then we are done.
(3) If we are not done, pick a vertex v' on the path above, and repeat 2. with v'.
(4) Piece together all the paths to get an Eulerian cycle.

Test out the algorithm on a few of the sample graphs. Try to break the algorithm!

a) In step 2., how do we know we that following (arbitrary edges) from v will eventually lead back to v?

b) After step 2. (i.e. removing the edges we've already used), are we left with a single graph or multiple graphs? Do these graphs have Eulerian cycles? Can you prove this?

5) a) Using the ideas from the algorithm in 4), finish your proof of the rule in 2b). Hint: You will probably want to use a form of induction (ask if you need a refresher!).

b) Adapt your proof in 5a) to prove that the algorithm in 4) always finds an Eulerian cycle in a graph if one exists.

6) Come up with an algorithm for finding Eulerian paths. Hint: Use the same type of trick as in 3b).

7) (Optional): Note our algorithm(s) above, in some sense, combine paths to eventually form the final Eulerian cycle/path. Try to come up with an algorithm that builds the path one edge at a time (never needing to go back and combine paths). Is this possible? What difficulties do you face? Hint: Think about when following an edge is a BAD idea.

Directed Graph Case

∞) Using the directed graphs on the graph examples page "repeat the above problems". Note: Part of this question is rewording the problems above as needed so they make sense in the directed graph cases.

Hint 1) Instead of degrees of vertices, examine both in-degrees and out-degrees.

Hint 2) We now have to worry about (maybe?) strongly-connected and weakly-connected graphs.