Section 6.2: Euler Paths and Hamiltonian Circuits

March 16, 2005

Abstract

Graphs are useful for characterizing two interesting problems: the traveling salesman problem, and the highway inspector problem. The problem in each case is to traverse a network in an optimal way, whether the focus is on the nodes (Sir William Rowan Hamilton, 1805-1865) or the arcs (Leonhard Euler, 1707-1783).

1 Euler Paths (the Highway Inspector problem)

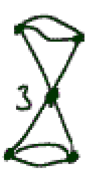
Definition: an **Euler Path** is a path in which each arc is used exactly once.

Euler got interested in these arcs when he encountered the Königsberg bridge problem (p. 425); a game in which the object was to cross every bridge without crossing any bridge twice. Euler solved this problem by inventing and then using Graph Theory, basically!

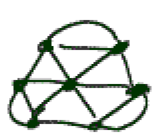
The bridges are the arcs, and the land masses are nodes, turned into the graph of Figure 6.5, p. 426.

To do: Practice 7, p. 426

4 odd
1 even

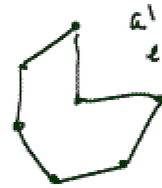


a.



b.

all even



all even

two odd

Theorem: in any graph, the number of odd nodes (nodes of odd degree) is even.

Outline of author's proof:

1. Suppose that there are A arcs, and N nodes. Each arc contributes 2 ends; the number of ends is $2A$, and the degrees d_i satisfy

$$2A = \sum_{i=1}^{N} d_i$$

2.

$$2m = 2A - \sum_{i|d_i \ even} d_i = \sum_{i|d_i \ odd} d_i$$

and the left hand side is even (call it $2m$).

3. The sum of two odd degrees is even, so assume (we proceed by contradiction) that there is an odd number of odd nodes. We can pair up all but one (say $i = k$), and then

$$\sum_{i|d_i \ odd;\ i \neq k} d_i = 2n$$

4. From which we conclude that

$$2m - 2n = d_i$$

which means that d_i was, in fact, even; but this is a contradiction. Hence, the number of odd nodes is even.

Alternate proof: by induction on number of arcs, using cases.

Theorem: an Euler path exists in a connected graph $\iff$ there are either two or zero odd nodes.

- Is this obvious? Why only two odd nodes?

- The two odd case reduces to the even case: start at one odd node, and trace a path to the other. Remove this subgraph, and what's left (and what might that be?) has even nodes only; so, since an Euler path exists for even noded graphs, we can reattach the pieces to form the original graph, with its Euler path.

To do: Practice 9, p. 428: Is the Königsberg bridge walk possible?

2 EulerPath Algorithm

The EulerPath algorithm (p. 428) makes use of the adjacency matrix representation of a graph to check for Euler paths. It simply counts up elements in a row i of the matrix (the degree of node i), and checks whether that's even or odd; if in the end there are not zero or two even nodes, there's no Euler path!

Problem: There's some worry about loops. How could we alleviate the problem with the smallest change in the code?

To do: Exercise 12, p. 432

EulerPath is $O(n^2)$, meaning that the number of operations in the worst case is on the order of n^2.

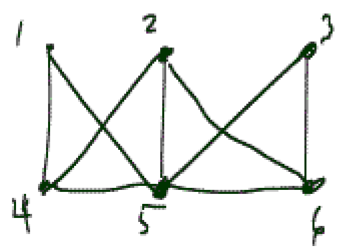
3 Hamiltonian Circuit Problem (the traveling salesman problem)

Definition: a **Hamiltonian Circuit (or Cycle)** is a cycle using every node of the graph (as a cycle, no node but the first is ever revisited).

"For example, consider a robot arm assigned to solder all the connections on a printed circuit board. The shortest tour that visits each solder point exactly once defines the most efficient path for the robot. A similar application arises in minimizing the amount of time taken by a graphics plotter to draw a given figure." (from http://www.cs.sunysb.edu/tildaroooniealgorith/files/traveling-salesman.shtml)

#12

$$\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}
\end{array}$$



An example is a complete graph, like K_5: there is a path from each node to every other node, so no matter where you start, you can trace a cycle through every node (breaks down for K_2!).

To do: Practice 11, p. 430

To do: Exercise 15, p. 432 (using trees, symmetry, and exhaustion!)

Unfortunately, there's no nice HamiltonCircuit algorithm for determining when there is a circuit (only very grungy, computationally intensive ones!). The traveling salesman problem (the optimal Hamiltonian Circuit on a weighted graph) is the poster child for the NP-complete problem (see p. 589, if interested!).

Alternatively, by induction on the number of arcs & by cases.

Base case: $A = 1$

Two ways to use the arc:

1. loop

2. connect two distinct nodes:

we now have an even # of odd nodes.

Inductive Step: assume that it's true for

Graphs of k arcs: they have an even # of odd nodes.

Introduce an arc:

1. loop, to node n_i

   degree(n_i) goes to degree(n_i) + 2

   (doesn't change even or oddness)

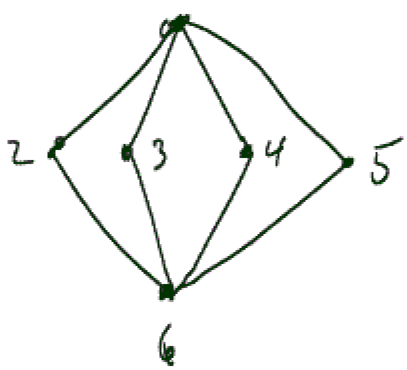
2. Connect two even nodes, in which case they both become odd as a pair

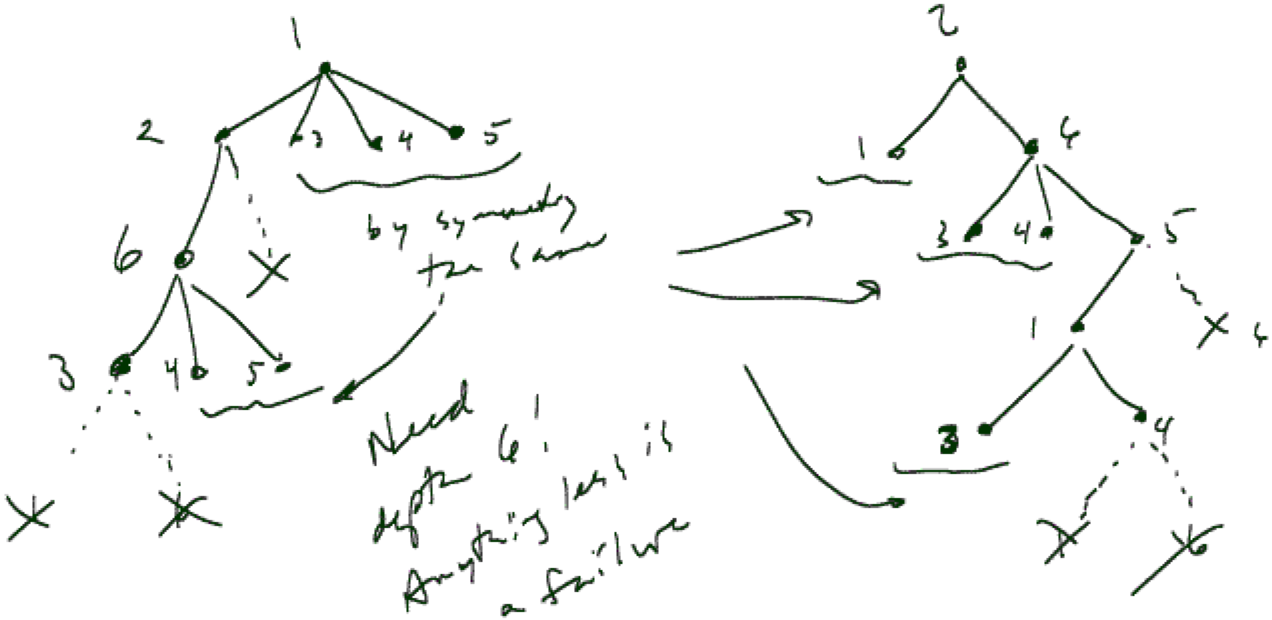
3. Connect two odd nodes, in which case we lose a pair of odd nodes.

4. Connect one of each, then they just switch parity; you still have the same number of odd nodes.

✓

#15, refering to #2





Proof by exhaustion
(using symmetry)