Finding Eulerian tours in linear time

Angus Lim
CPSC 490

February 6, 2015

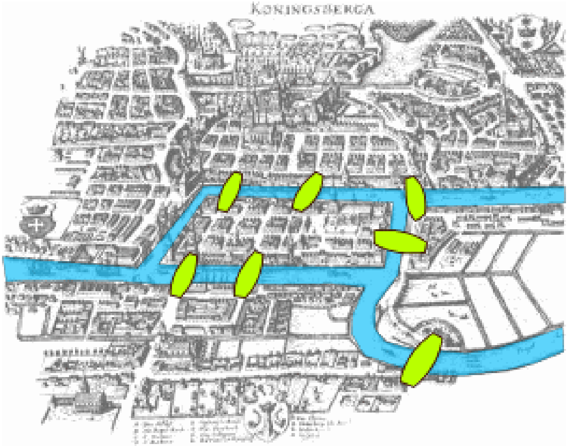
What is an Eulerian tour anyway

I'm sure this is review

- aka Eulerian cycle/Eulerian circuit
- directed path on graph
  - visits every edge *once*
  - starts and ends at the *same* vertex
- cf. Eulerian paths, which do not have the second condition
- we'll work with connected, undirected graphs because they're easier and Paul didn't specify what kind of graph

Historical context

totally not ripped off Wikipedia



Finding whether a tour even exists

a formal proof is "outside the scope of this course"

- consider an arbitrary vertex v
  - in order for an Eulerian tour to exist, every edge adjacent to v must be visited
  - for every edge *leaving* v, there must be an edge *returning* to v
  - if there are n edges leaving v, there are n edges returning to v, a total of $2n$ edges
  - since n is an integer, there must be an *even* number of edges adjacent to v
- this is true for all vertices
- ergo, a necessary condition for existence of an Eulerian tour is that *every vertex must have even degree*
- this can be evaluated in $\mathcal{O}(|E|)$ time

Can we prove the converse?

- i.e., if we have a graph where every vertex has even degree, can we find an Eulerian tour?
- this is equivalent to the original problem
- we present the following algorithm, published by Carl Hierholzer in 1873

Finding the actual tour: Hierholzer's algorithm

throwback to 1873

- using what we figured out earlier, if we leave an arbitrary vertex *v* during path traversal, we are guaranteed that we can return to it
- we can thus do the following recursion, while there are unvisited edges:
  - start at a vertex *u*, and arbitrarily traverse unvisited edges until *u* is reached again
  - this creates a subtour, which may not traverse all edges
  - recurse on vertices in the found subtour; if there are unvisted edges adjacent to it, we'll get another subtour, which can be inserted into the full tour
- using doubly linked lists, the runtime is $\mathcal{O}(|E|)$

An example

- note to self: you were too lazy to make nice diagrams in Inkscape so use the chalkboard

Questions?