Summary on Lecture 8, January 20, 2016

Introduction to Graph Theory: more.

Let n be a positive integer. Then a *complete graph* K_n is a graph with n vertices $v_1, \dots, v_n$ and $\binom{n}{2}$ edges $e_{i,j} = \{v_1, v_j\}$, where $i \neq j$.

A complete graph K_n, contains subgraphs isomorphic to the graphs K_m for $m = 1, 2, \dots, n$. Such a subgraph can be obtained by selecting any m of the n vertices and using all the edges in K_n joining them. Thus K_5 contains $\binom{5}{2} = 10$ subgraphs isomorphic to K_2, $\binom{5}{3} = 10$ subgraphs isomorphic to K_3 [i.e., triangles], and $\binom{5}{4} = 5$ subgraphs isomorphic to K_4. In fact, every graph with n or fewer vertices and with no loops or parallel edges is isomorphic to a subgraph of K_n; just delete the unneeded edges from K_n.

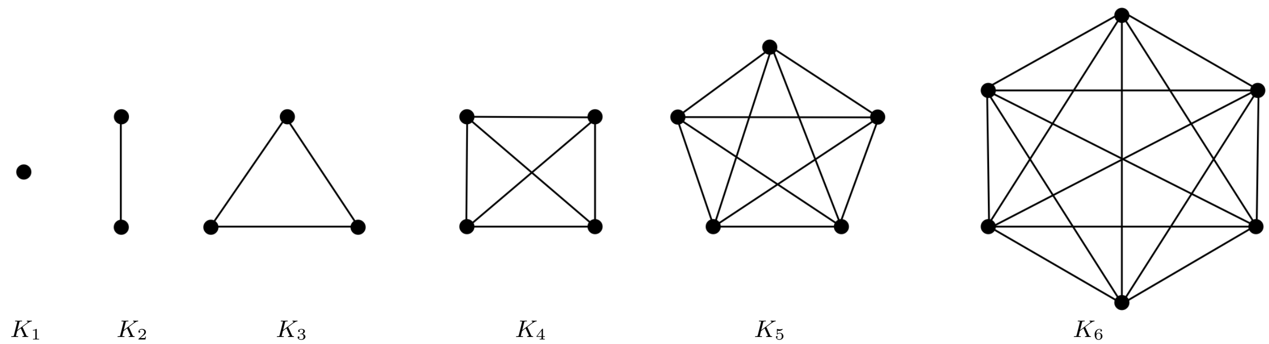

Fig. 6. Complete graphs

Complete graphs have a high degree of symmetry. Each permutation α of the vertices of a complete graph gives an isomorphism of the graph onto itself, since both $\{u, v\}$ and $\{(\alpha(u), \alpha(v)\}$ are edges whenever $u \neq v$. The next theorem relates the degrees of vertices to the number of edges of the graph.

Theorem 2. *The sum of the degrees of the vertices of a graph* $G = (V, G)$ *is twice the number of edges, i.e.,*

$$\sum_{v \in V} \deg(v) = 2 \cdot |E(G)|.$$

Proof. Each edge, whether a loop or not, contributes 2 to the degree sum. This is a place where our convention that each loop contributes 2 to the degree of a vertex pays off. $\square$.

Euler Trails and circuits

The Seven Bridges of Königsberg. *The Seven Bridges of Königsberg Problem* is a historically important problem in mathematics. Its negative resolution by Leonhard Euler in 1736 laid the foundations of *graph theory* and prefigured the idea of *topology*.

Fig. 7. The Seven Bridges of Königsberg[1]

Here is The Seven Bridges of Königsberg Problem: find a walk through the city that would cross each bridge once and only once, with the conditions that the islands could only be reached by the bridges and every bridge once

[1] These pictures are taken from Wikipedia

accessed must be crossed to its other end.

Since only the connection information is relevant, the shape of pictorial representations of a graph may be distorted in any way, without changing the graph itself. Thus it is enough to analyze the corresponding graph (on the left of Fig. 7). A closed walk which uses each edge only once is called an *Euler circuit*.

A key observation due to Euler is that whenever one enters a vertex by a bridge, one leaves the vertex by a bridge. In our terms, it means that if a graph has an Euler circuit, then a degree of every vertex has to be even. It sounds too easy, however, there is a remarkable result that this is the only condition for existence of an Euler circuit:

Theorem 3. (Leonard Euler, 1736)
Let G be a finite connected graph. Then G has an Euler circuit if and only if all vertices of G have even degrees.

We prove Theorem 3 later. We say that a walk in a graph G is an *Euler trail* if it uses every edge only once.

Corollary 4. *Let G be a finite connected graph. Then G has an Euler trail if and only if it has either two vertices of odd degree or no vertices of odd degree.*

Proof. Suppose that G has an Euler trail starting at v and ending at v'. If $v = v'$, the path is closed and Theorem 3 says that all vertices have even degree. If $v \neq v'$, we create a new edge e joining v and v'. The new graph $G \cup \{e\}$ has an Euler circuit consisting of the Euler trail for G followed by e, so all vertices of $G \cup \{e\}$ have even degree. Then we remove the edge e. Then v and v' are the only vertices of $G = (G \cup \{e\}) \setminus \{e\}$ of odd degree. □

Remark. Returning to The Seven Bridges of Königsberg Problem, we see that there is no an Euler trail for the graph from Fig. 7. Indeed, all four vertices have odd degree.

Finding an Euler Circuit

In order to prove Theorem 3, we would like to describe an algorithm how to find an Euler circuit if all vertices of G have even degrees. We start with an algorithm which finds a circuit which is not necessarily an Euler circuit, i.e. it may visit only once some of edges.

Let $H = (V(H), E(H))$ be a graph with all verices of even degree and let $v \in V(H)$ be a vertex with positive even degree. For a graph G and an edge e, we define a graph $G \setminus \{e\}$ which has exactly the same vertices as G and the same edges except given edge e. We say that the graph $G \setminus \{e\}$ is given by removing e from $E(G)$. Here is the algorithm:

Circuit (H, v)

```
Choose an edge e with endpoint v
Let P := (e) and remove e from E(H)
while there is an edge at the terminal vertex of P do
      Choose such an edge e and add it to the path:
      P := (P, e) and remove it from E(H),
return P
```

Exercise: Analyze the algorithm **Circuit** (H, v). Why does it produce a circuit?

Now we are ready for an algorithm which produces and Euler circuit.

EulerCircuit $G = (V, E)$ ($\deg v$ is even for each $v \in V$)

```
Choose a vertex v ∈ V(G)
Let C := Circuit(G, v)
while length(C) < E(G) do
      Choose a vertex w in C of positive degree in G \ C.
      Attach Circuit(G \ C, w) to C at w to obtain a longer circuit C.
return C
```

Exercise: Analyze the algorithm **EulerCircuit** (G). Why does it produce an Euler circuit?