Notes

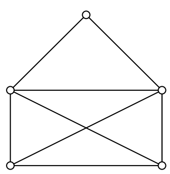
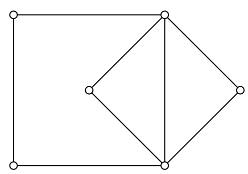
4 fevrier, 2015

3.1 eulerian paths

if G is a non-trivial graph containing a path that visits every edge then we call the path Eulerian. If the trail is closed, (a cycle) then we call the graph Eulerian.

can you draw it without picking up pencil?

example

eulerian path:  eulerian graph 

theorem

let G be a connected (not necessarily simple graph). G is Eulerian if and only if $d(v)$ degree of v is even $\forall v \in V(G)$

proof

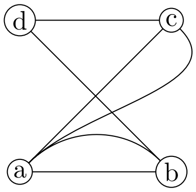
forward direction:

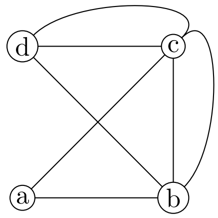
let C be an eulerian cycle. C visits every edge which means each time it enters a vertex, it also exits. except first and last? the degree increases by 2 for each visit. increase of one when we start, increase of 1 when finishing for the starting/endpoint. so the source vertex has even degree also.

reverse direction:

we are given that every vertex is even degree. let P be a $u-v$ path for some $u, v \in V(G)$. However, since $\deg(v) = 2k$ we have an exit available. choose an available neighbor v' and extend P to include v'. Now P is a $u-v'$ path. but $\deg(v') = 2k'$ and so we have an exit. repeat and eventually we will hit u again ($|V(G)| < \infty$) creating a cycle. Call this cycle C and assume it is not eulerian, else done. let $H = G - C$. remove edges in C and the hanging vertices. let $x \in V(C)$ that is adjacent to and edge not in C. then this point is in H and exists because C is not eulerian. let H' be a component of H such that $x \in V(H')$. the degrees of all vertices in H' are necessarily even because C removed an even number of edges from each vertex. Repeating, we can create an $x-x$ path in H'. If the $x-x$ cycle is not eulerian in H', recurse (graph is finite). the cycle is: start on C when we get to another cycle, traverse it, and then continue on with C.

bridges of königsberg:

old


new


corollary

let G be a connected graph, G has an eulerian trail if and only if G has two or fewer odd vertices.

proof

$\Leftarrow$ if 0 odd vertices we are done. if G has two odd vertices, add an edge between them, create the E.C. and delete the edge.

$\Rightarrow$ "go in, go out" argument.

homework

3,4,6,7