MCS-236: Graph Theory Handout #Ch1
San Skulrattanakulchai
Gustavus Adolphus College
Sep 17, 2012

Chapter 1. Introduction

Definition. *Let*

$$X : x = v_0, v_1, \ldots, v_{k-1}, v_k = y$$

be an $x - y$ walk and let

$$Y : y = v_k, v_{k+1}, \ldots, v_{k+\ell-1}, v_{k+\ell} = z$$

be a $y - z$ walk. We say that the walk $Z = v_0, \ldots, v_k, v_{k+1}, \ldots, v_{k+\ell}$ results from concatenating Y to X.

Let X be as above and let i and j be such that $0 \le i < j \le k$. Then the $v_i - v_j$ walk $X' : v_i, v_{i+1}, \ldots, v_{j-1}, v_j$ is said to be a subwalk *of X.* Deleting the subwalk X' from X *means "removing all edges and internal vertices of X' from X." If $v_i \neq v_j$, then we get two distinct walks (a $v_0 - v_i$ walk and a $v_j - v_k$ walk) after deletion. But if $v_i = v_j$, then we get one walk (a $v_0 - v_k$ walk) after deletion.*

We prove Theorem 1.6 by algorithm.

Theorem (Theorem 1.6, CZ)**.** *If a graph G contains a $u - v$ walk of length ℓ, then G contains a $u - v$ path of length at most ℓ.*

Proof. Given a $u - v$ walk W in G of length ℓ, we execute the following algorithm.

```
while W contains repeated vertices do {
   let x be some vertex that occurs (at least) twice on W
   delete an x − x subwalk from W
}
return W as the desired path
```

We prove this algorithm correct by showing that

1. If the algorithm terminates, then it returns a $u - v$ path of length at most ℓ.

2. The algorithm terminates.

First note that W is a $u - v$ walk before and after each iteration of the while loop. Suppose the algorithm terminates. Then line 5 must have been executed, which means the while loop exits. Since the loop exits only when W contains no repeated vertices, we see that the algorithm returns a $u - v$ path. This path must have length at most ℓ since it is derived from the input walk by having some (if any) subwalk(s) deleted from it.

Each time the body of the loop executes, the length of the walk W decreases by some positive amount. Now, a walk of shortest possible length is the trivial walk of length 0. Since the input walk has length ℓ, the while statement iterates no more than ℓ times. This means the algorithm terminates.

This completes the proof. $\square$

Definition. *Let $G = (V, E)$ be a graph. A path $P : v_0, v_1, \ldots, v_{\ell-1}, v_\ell$ in G is called* maximal *if all neighbors of the ends of P are on P. In other words, if v_0x is an edge of G then $x = v_i$ for some $0 < i \leq \ell$, and if $v_\ell x$ is an edge of G then $x = v_i$ for some $0 \leq i < \ell$.*

Exercise. Give an algorithm for getting a maximal path.

We prove Thereom 1.9 by consideration of a maximal path (instead of longest geodesic like in CZ).

Theorem (Theorem 1.9, CZ)**.** *If G is a connected graph of order 2 or more, then G contains two distinct vertices u and v such that $G - u$ is connected and $G - v$ is connected.*

Proof. Let P be a maximal path in G, and let u and v be the end vertices of P. Since G is connected and nontrivial, G has no isolated vertex. Thus, none of G's maximal paths is trivial. Therefore, $u \neq v$.

First we'll prove that $G - u$ is connected. Let x and y be any vertices in $G - u$. Since G is connnected, G contains an $x - y$ path, say Q. We consider two possibilities.

Case 1: u is not on Q. Then Q is an $x - y$ path in $G - u$ as well. Thus, $x \sim y$ in $G - u$.

Case 2: u is on Q. Then u appears on Q exactly once since Q is a path. Say that $Q : x = w_0, w_1, \ldots, w_i, w_{i+1} = u, w_{i+2}, \ldots, w_k = y$. Since P is a maximal path with u as one of its end vertices, all neighbors of u is on P. This means that P contains a $w_i - w_{i+2}$ subpath P'. Now let Q' be the result of replacing the subpath w_i, u, w_{i+2} on Q by P'.

Then Q' is an $x - y$ walk in $G - u$. By Thereom 1.6, Q' contains an $x - y$ path (in $G - u$). Thus, $x \sim y$ in $G - u$.

In both cases, we have shown that $x \sim y$ in $G - u$. Thus, $G - u$ is connected.

That $G - v$ is connected can be proved in a similar way. $\square$