(Let us complete the proof from last lecture)

P

↓

(a longest path in G)

Let u & v denote the ends of P.

(Now, how do we complete the proof?)

Observe that, in the subgraph P,

u & v are vertices of degree one.

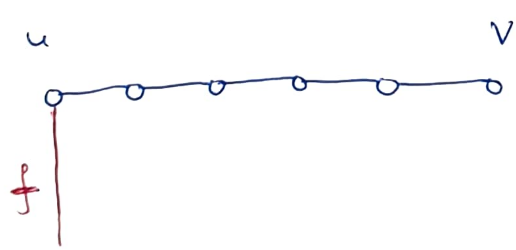
But, in G, each vertex has degree

at least two.

So, u must have another edge

incident, say f, such that $f \notin E(P)$.

P



→ Let us check/debug the proof:

→ What if $u = v$?

↓

when is this possible? Remember P is a longest path.

↓

$u = v$ is possible ONLY IF each edge of G is a loop (why?)

↓

How do we deal with this? (Remember: any loop is a cycle.)

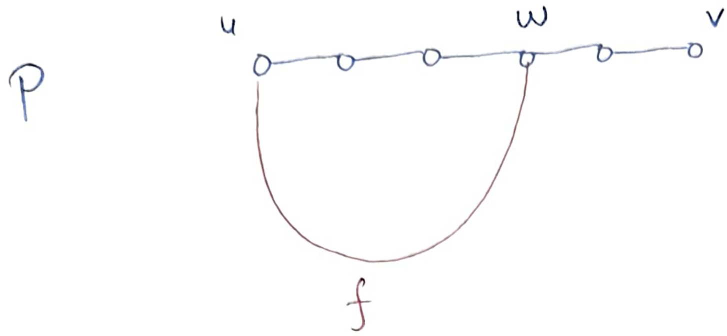
u is one end of f. Where is the other end of f?

Let w denote the other end of f.

If $w \notin V(P)$ then $E(P) \cup \{f\}$ is a longer path;

contradiction (since P is a longest path).

So, $w \in V(P)$.



Now, observe that we have a cycle:

Consider the subpath of P from u to w and edge f.

This is a cycle. (Right?) $\boxed{\diagup}$

Checking/Debugging continued:

One possibility:

At beginning of proof:

If G has a loop then G has a cycle. NOTHING TO PROVE.

Now suppose that G is loopless.

So, we have finished our first nontrivial proof (of the following theorem):

An exercise in logic

Theorem: Let G be a graph.
If each vertex of G has degree at least two then G is NOT a forest.

↕ same as

Theorem: Let G be a graph.
If G is a forest then G has at least one vertex whose degree is at most one.

DIY: Convince yourself that these two statements have the same meaning.

<u>Question</u>: How does one discover such a proof?

↓ easier question

<u>Question</u>: How would anyone have discovered <u>this</u> proof?

↓ let us try to answer this

<u>Hypothesis</u>: G is a graph whose each vertex has degree is at least two

<u>Desired conclusion</u>: G has a cycle (as a subgraph).

Basically we are trying to "make" a long path. We will return to this point later.

↓

So we need to show existence of cycle.

↓

It seems natural to try to find/construct a cycle.

↓

Start at some vertex v; now "construct" a path starting at v — keep "walking" (and "mark" vertices you have already seen) until you return to a vertex that is already "marked"

↓

this will happen since graph is finite & each vertex has degree ≥ 2

This type of "algorithmic thinking" may lead one to the idea of taking a longest path