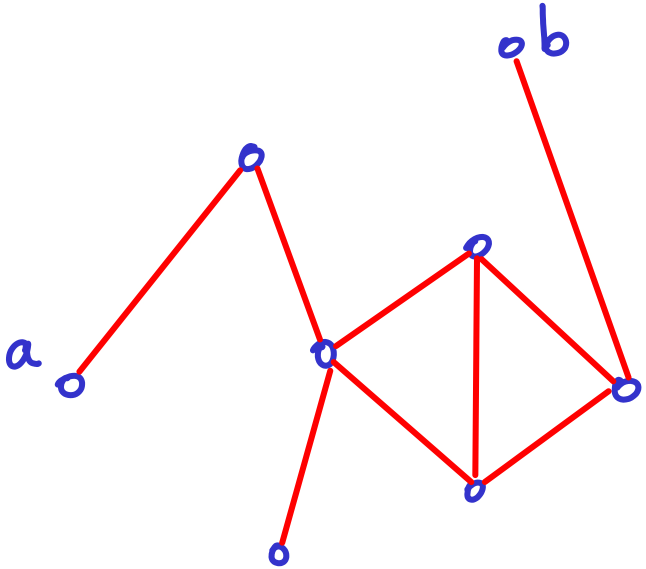
GRAPH CONNECTIVITY

GRAPH CONNECTIVITY



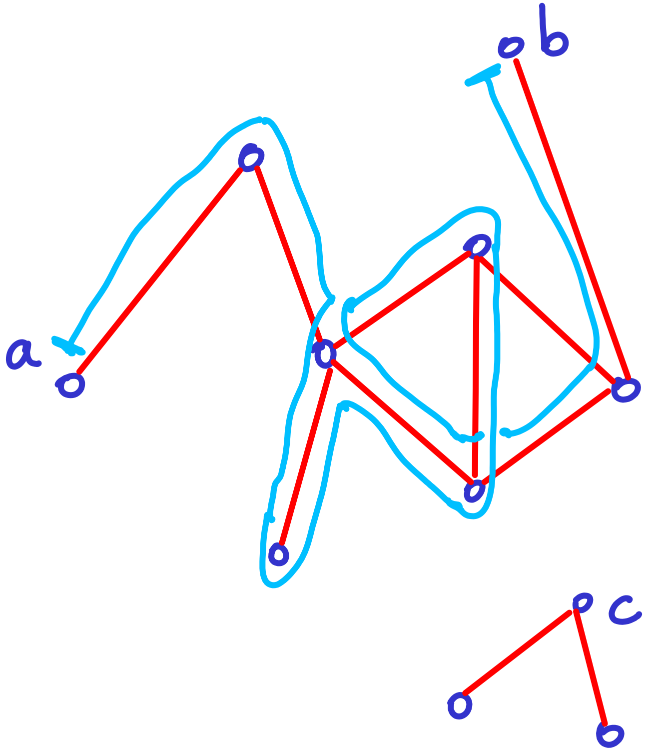
A **walk** is a sequence of vertices

$v_i, v_{i+1}, v_{i+2}, \ldots, v_k$

s.t. every v_j, v_{j+1} is an edge

$(i \leq j < k)$

GRAPH CONNECTIVITY

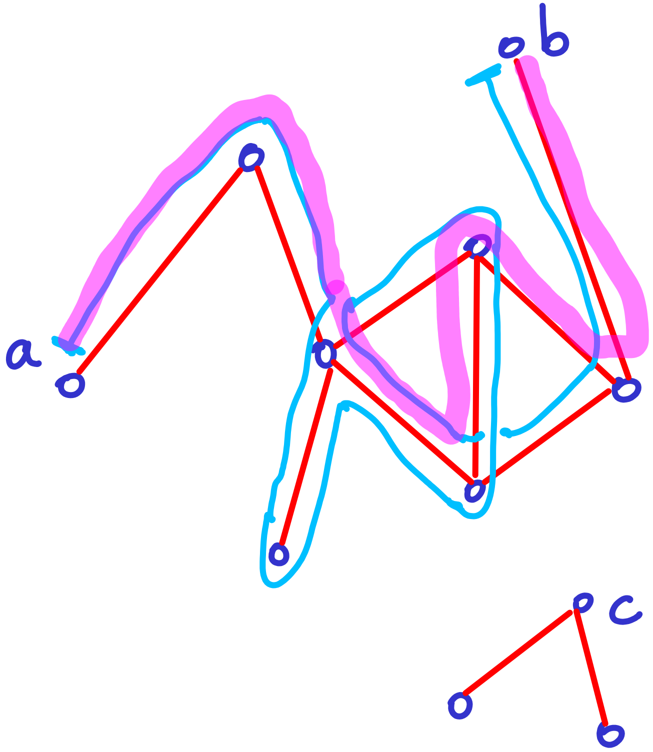


A **walk** is a sequence of vertices $v_i, v_{i+1}, v_{i+2}, \ldots, v_k$

s.t. every v_j, v_{j+1} is an edge $(i \le j < k)$

We can walk from a to b, but not from a to c.

GRAPH CONNECTIVITY



A **walk** is a sequence of vertices $v_i, v_{i+1}, v_{i+2}, \cdots, v_k$

s.t. every v_j, v_{j+1} is an edge
$(i \leq j < k)$

We can walk from a to b, but not from a to c.

A **path** is a walk with distinct vertices

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G) if there is a path from x to y

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G) if there is a path from x to y

could say "walk"

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G)

if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

[proof ?]

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G) if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G) if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

It must have a repeated vertex, u

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G) if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

It must have a repeated vertex, u } $x \ldots u \ldots u \ldots y$

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G)

if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

It must have a repeated vertex, u } $x \dots u \dots u \dots y$

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G)
if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

It must have a repeated vertex, u } $x \ldots u \underbrace{\ldots}_{\text{remove}} u \ldots y$

GRAPH CONNECTIVITY

A vertex x is connected to a vertex y (in G)

if there is a path from x to y

could say "walk"

If there is an x-y walk in G then there is an x-y path in G.

↳ Pick the shortest x-y walk that is not a path.

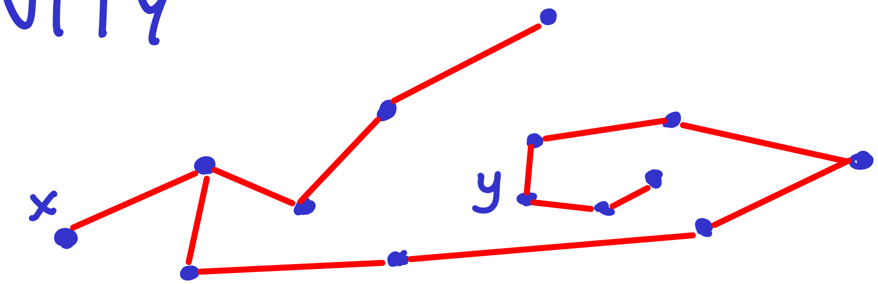
It must have a repeated vertex, u } $x \dots u \underbrace{\dots}_{\text{remove}} u \dots y$

CONTRADICTION

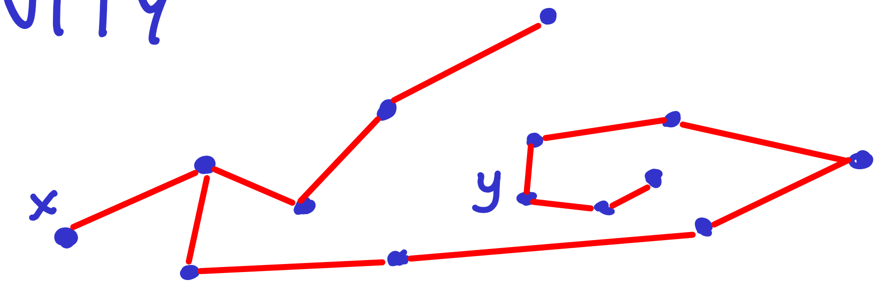
GRAPH CONNECTIVITY

A graph is connected

if every pair of vertices is connected



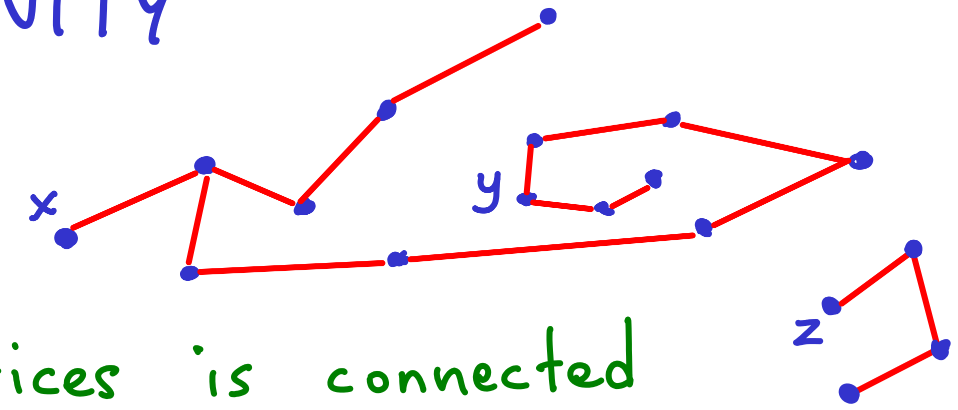
GRAPH CONNECTIVITY



A graph is <u>connected</u>
if every pair of vertices is connected

If vertex x is connected to vertex y
then they are in the same <u>component</u>.

GRAPH CONNECTIVITY



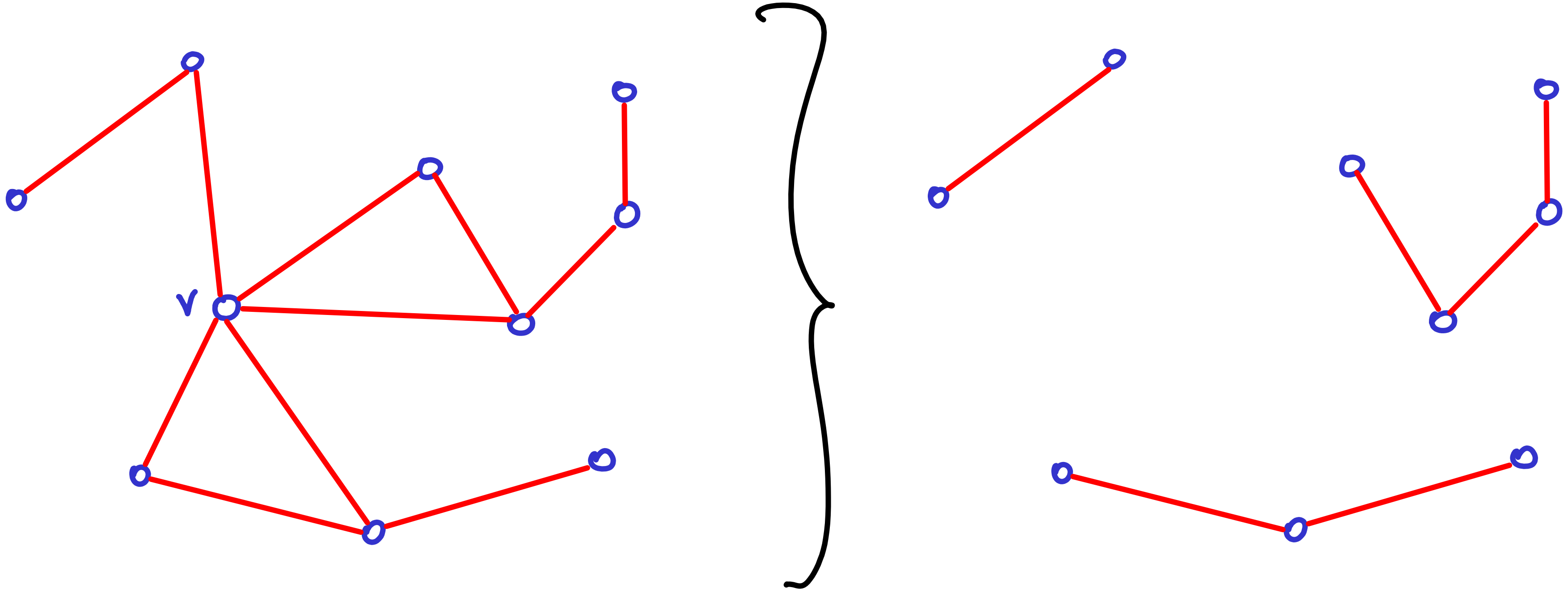
A graph is connected
if every pair of vertices is connected

If vertex x is connected to vertex y
then they are in the same component.

If x & y are in the same component
but x & z are not
then y & z are not.

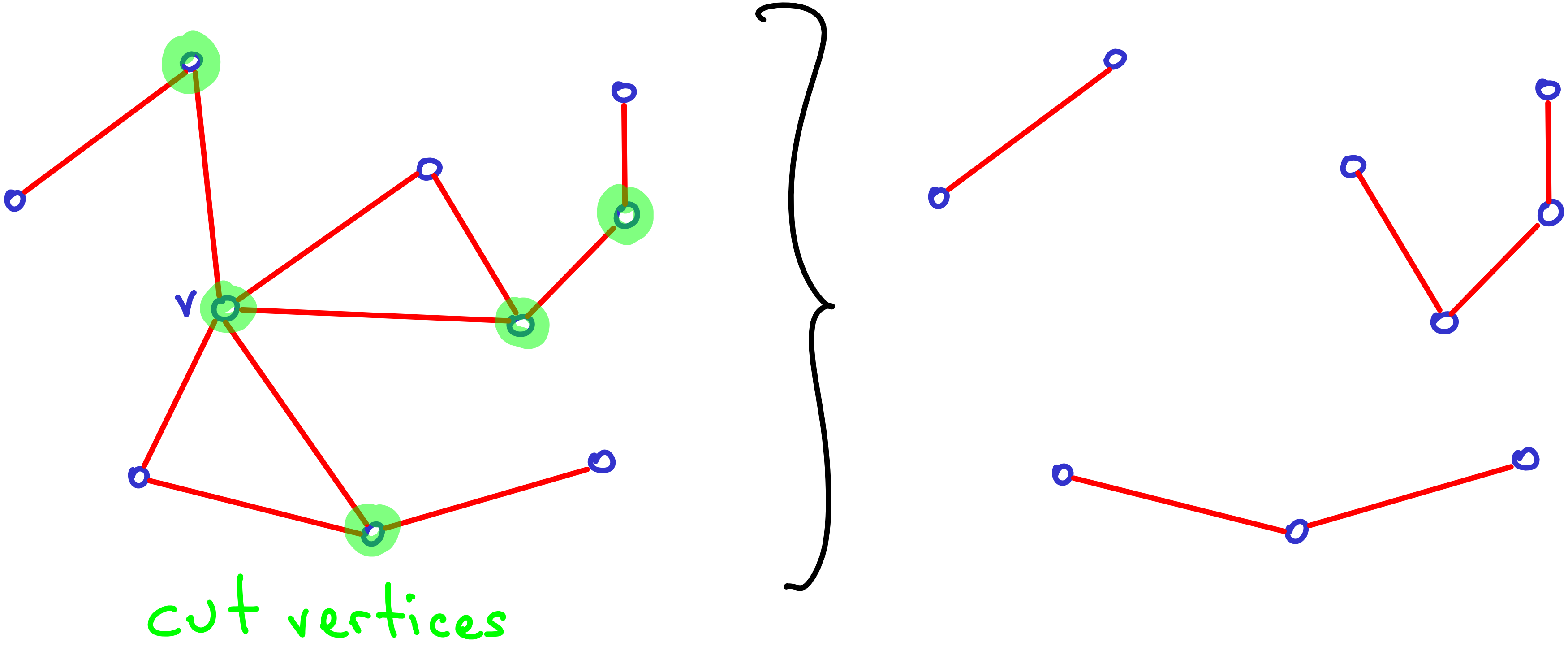
Given G, remove a vertex: $G-v$

If $G-v$ has more components than G, then v is a cut vertex.



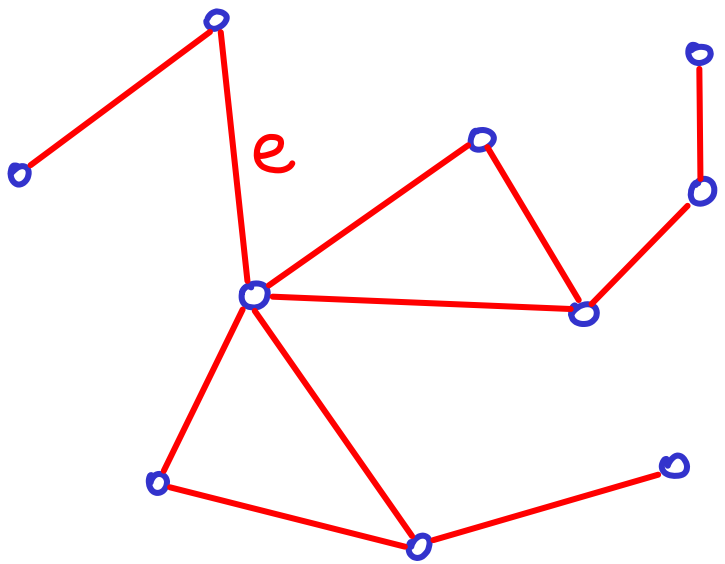
Given G, remove a vertex: $G-v$

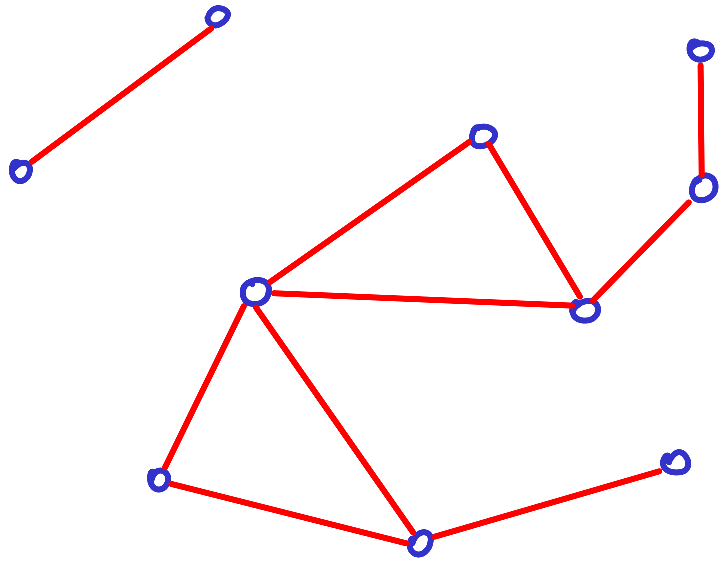
If $G-v$ has more components than G, then v is a cut vertex.



Given G, remove an edge : $G-e$

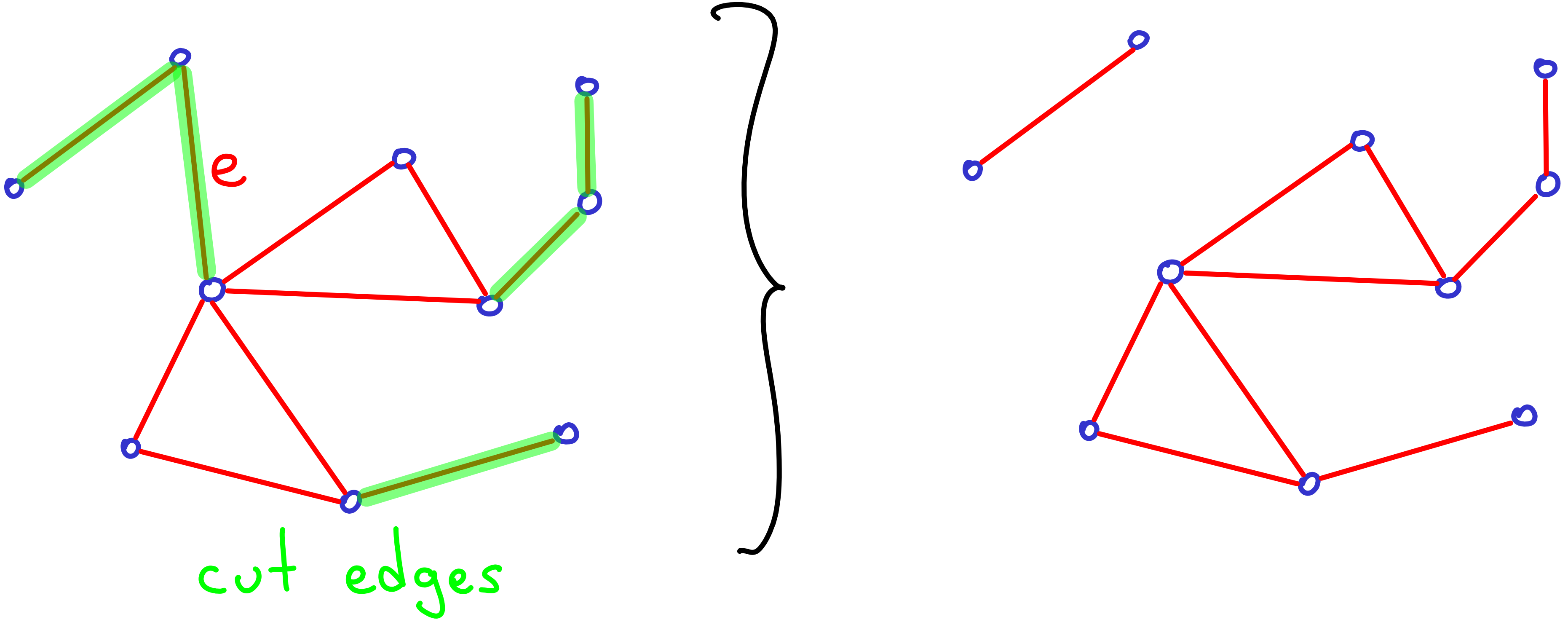
If $G-e$ has more components than G, then e is a cut edge.





Given G, remove an edge : $G-e$

If $G-e$ has more components than G, then e is a cut edge.

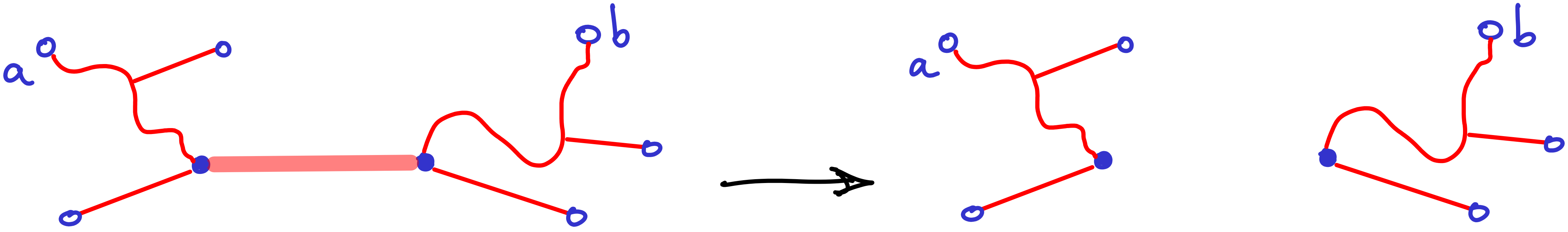


Claim: a cut edge can't be on a cycle.

(a cycle is a path w/ start = end)

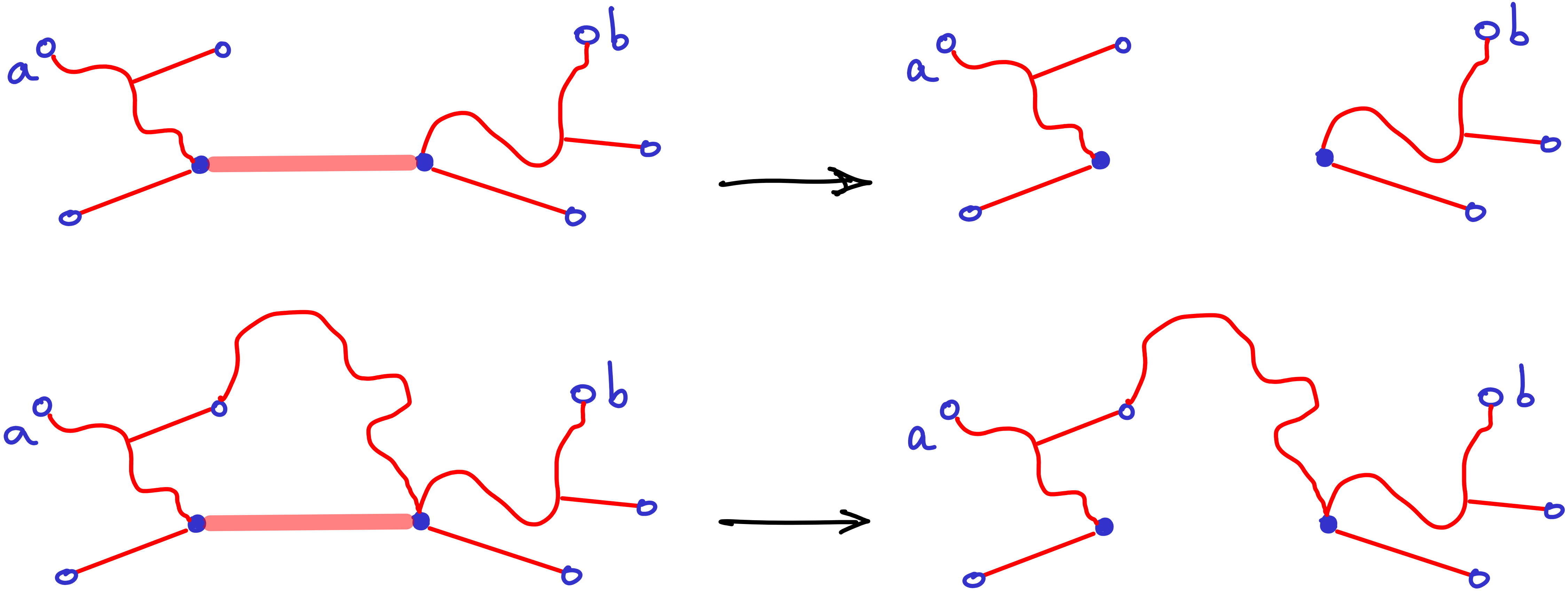
Claim: a cut edge can't be on a cycle.

(a cycle is a path w/ start = end)



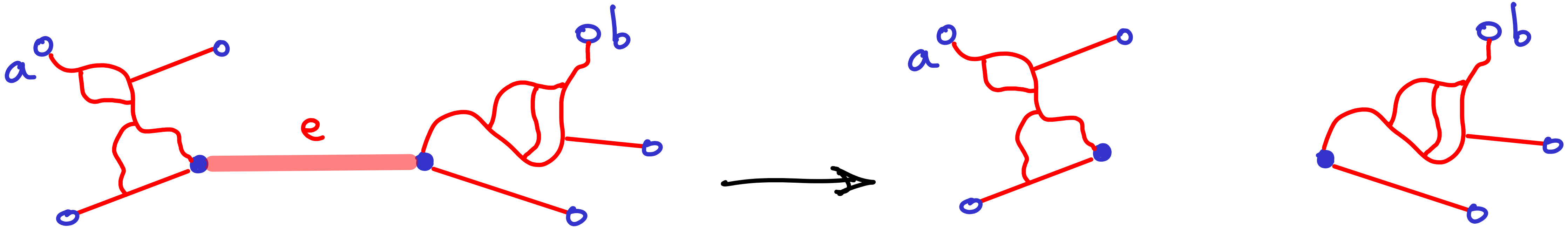
Claim: a cut edge can't be on a cycle.

(a cycle is a path w/ start = end)



... cont'd : if e is a cut edge then $\exists$ a, b s.t.

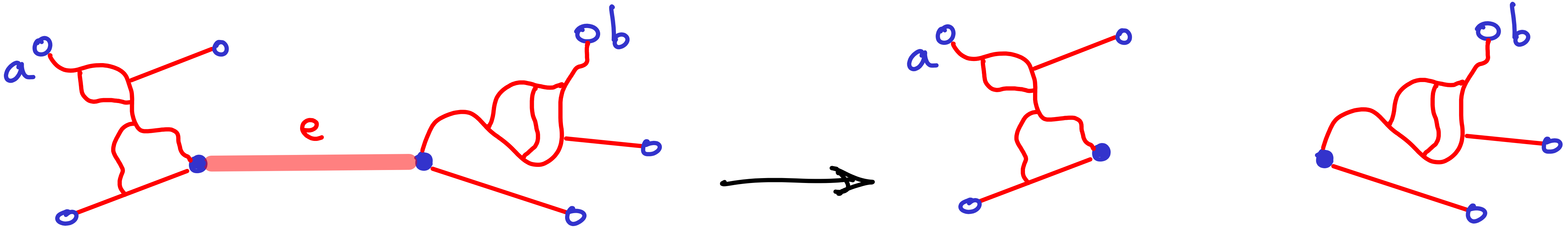
a is not connected to b in G − e.



...cont'd: if e is a cut edge then $\exists$ a, b s.t.

a is not connected to b in G − e.

So all paths from a to b in G use e

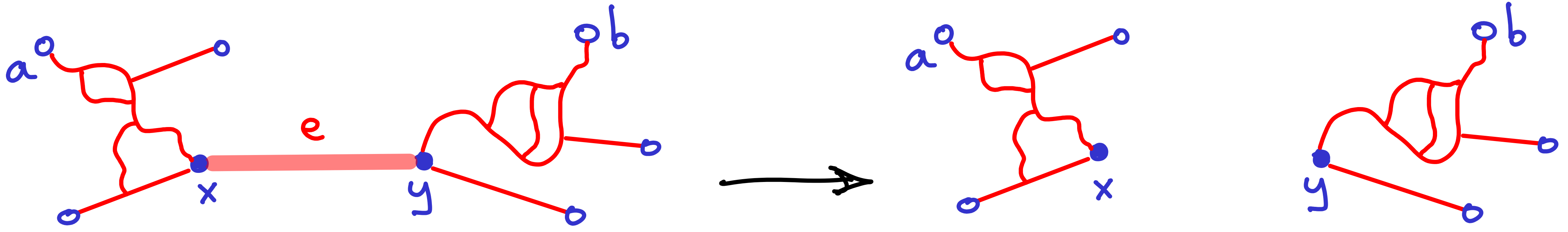


...cont'd: if e is a cut edge then $\exists\ a, b$ s.t.

a is not connected to b in $G - e$.

So all paths from a to b in G use e,

i.e traverse $a \cdots xy \cdots b$ where $e = \overline{xy}$.



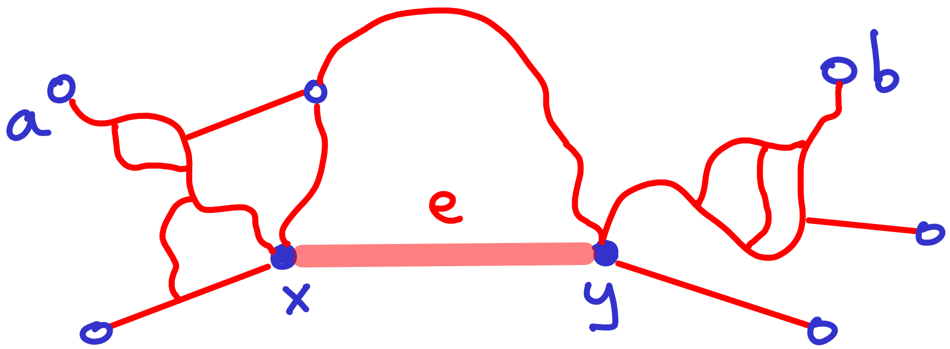
...cont'd : if e is a cut edge then $\exists\, a, b$ s.t.

a is not connected to b in $G - e$.

So all paths from a to b in G use e,

i.e traverse $a \cdots x y \cdots b$ where $e = \overline{xy}$.

But if e is on a cycle

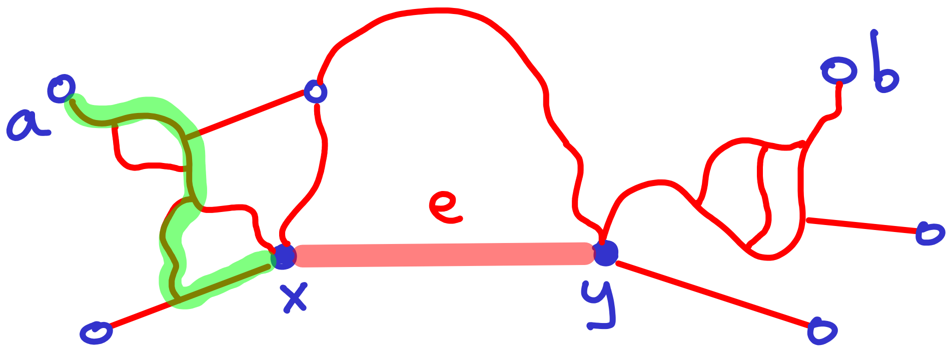


...cont'd : if e is a cut edge then $\exists\ a, b$ s.t.

a is not connected to b in $G-e$.

So all paths from a to b in G use e,

i.e traverse $a \cdots xy \cdots b$ where $e = \overline{xy}$.

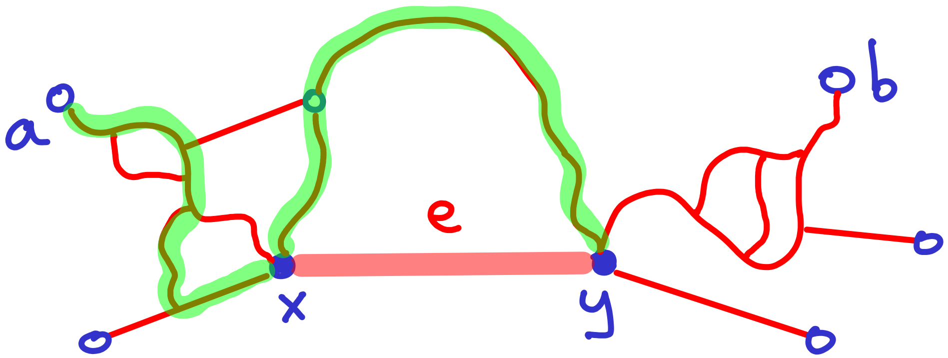


But if e is on a cycle

then we can walk from a to x,

...cont'd: if e is a cut edge then $\exists\ a, b$ s.t. a is not connected to b in $G - e$.

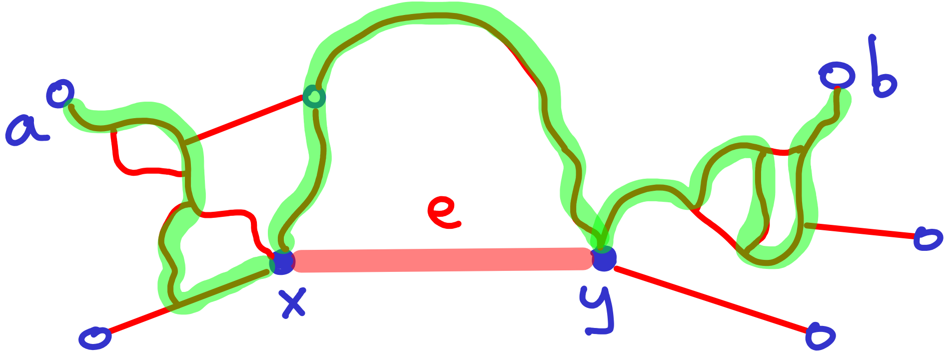
So all paths from a to b in G use e, i.e traverse $a \cdots xy \cdots b$ where $e = \overline{xy}$.



But if e is on a cycle then we can walk from a to x, walk from x to y, without using e,

...cont'd : if e is a cut edge then $\exists$ a, b s.t. a is not connected to b in $G-e$.

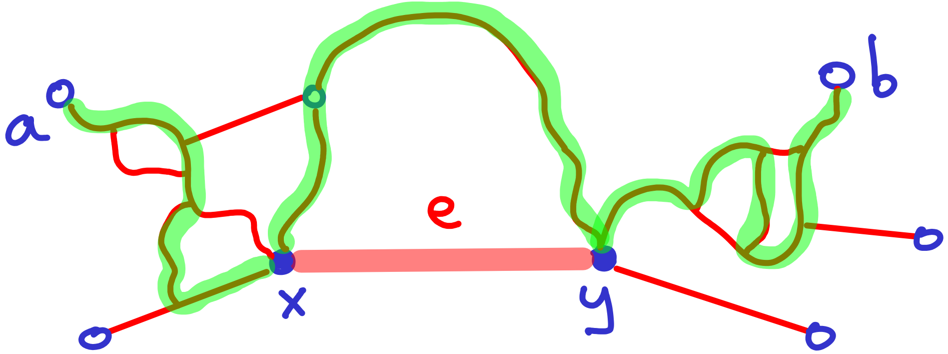
So all paths from a to b in G use e, i.e traverse $a \cdots xy \cdots b$ where $e = \overline{xy}$.



But if e is on a cycle then we can walk from a to x, walk from x to y, without using e, and walk from y to b.

...cont'd: if e is a cut edge then $\exists\, a, b$ s.t. a is not connected to b in $G-e$.

[So all paths from a to b in G use e, i.e traverse $a \cdots xy \cdots b$ where $e = \overline{xy}$.



But if e is on a cycle then we can walk from a to x, walk from x to y, without using e, and walk from y to b.

CONTRADICTION

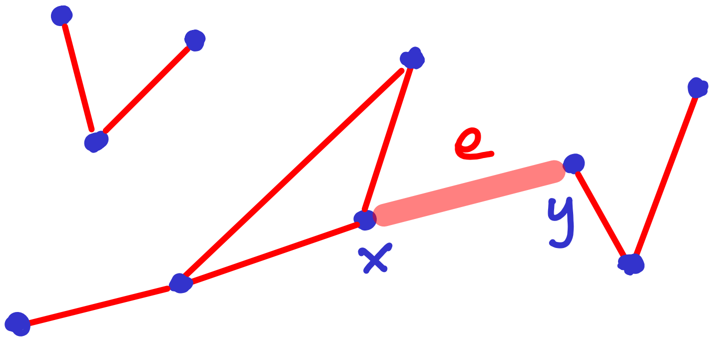
Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Types of vertex pairs (a, b) in G:

i) no path exists between a & b

ii) some path exists...

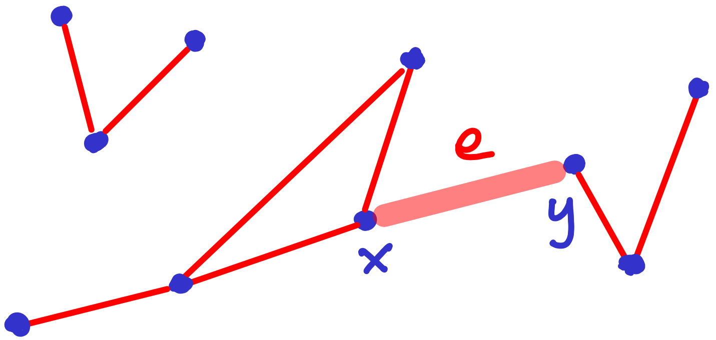


Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Types of vertex pairs (a, b) in G:

0) no path exists between a & b
1) not all paths between a & b use e
2) all paths use e

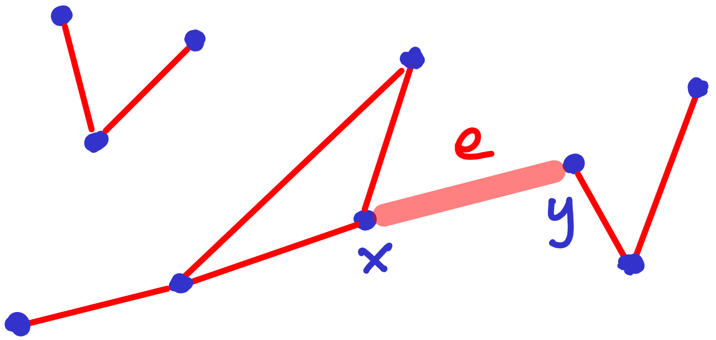
(path exists)



Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

Types of vertex pairs (a,b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e

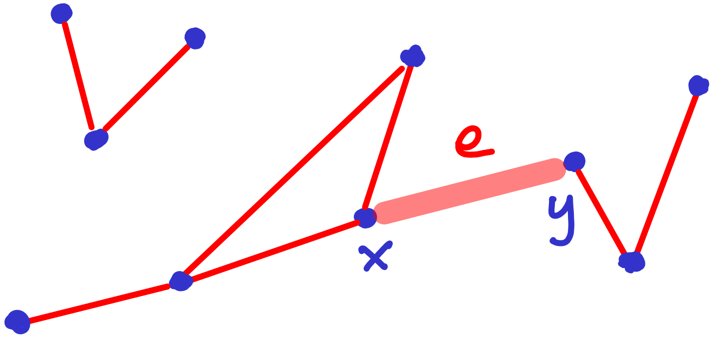


?

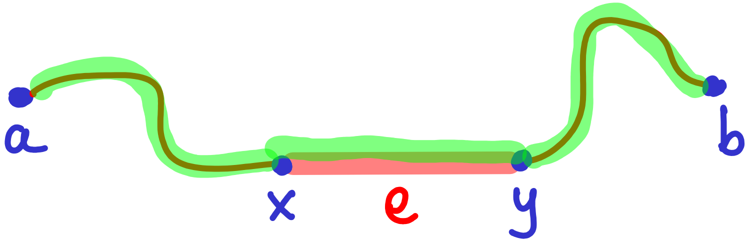
Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

Types of vertex pairs (a,b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e



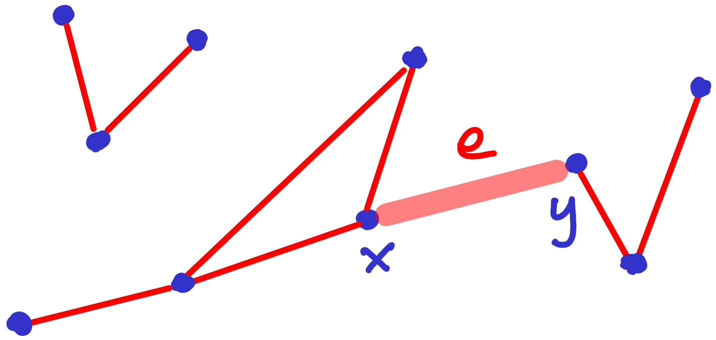
If path P uses e



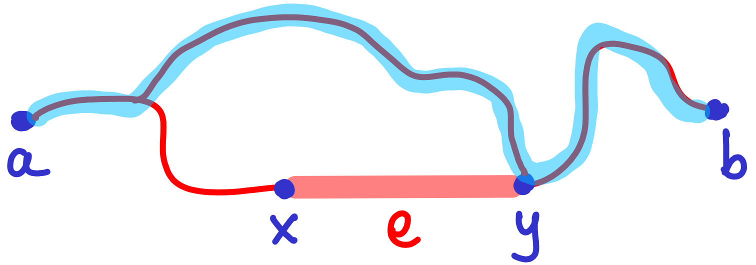
Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Types of vertex pairs (a, b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e



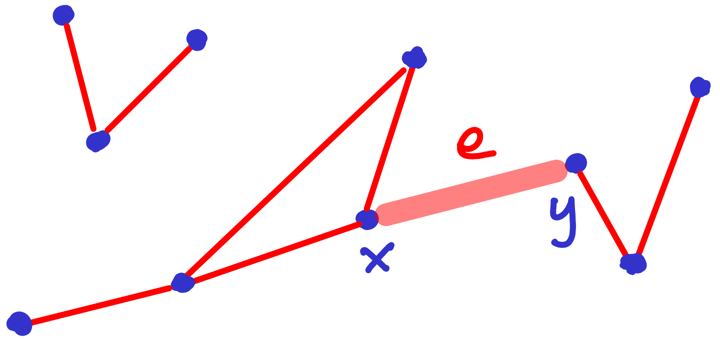
If path P uses e
& path Q doesn't



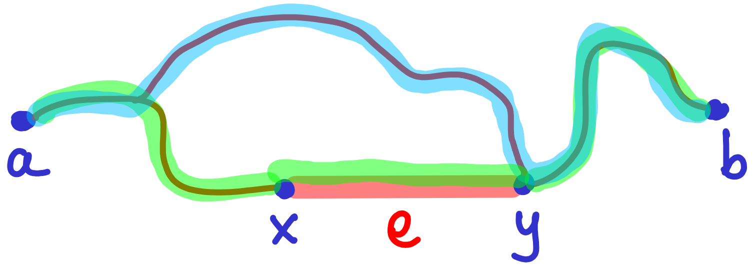
Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

Types of vertex pairs (a,b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e

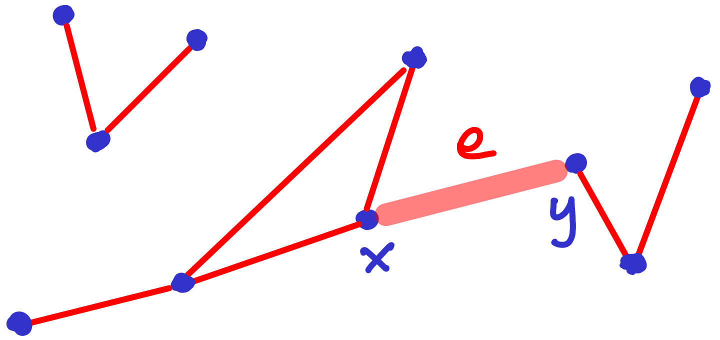


If path P uses e & path Q doesn't then e is on a cycle



Claim: Removing a cut edge e = (x, y) increases the number of components by 1.

Types of vertex pairs (a, b) in G:



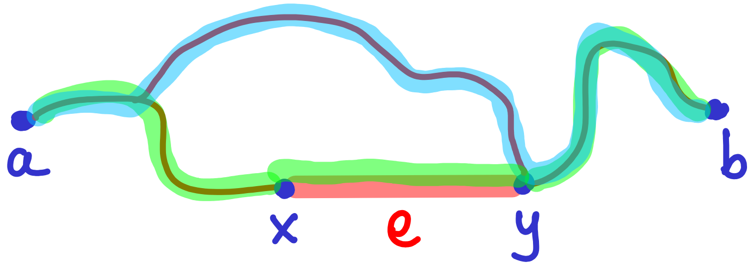
0) no path exists between a & b

1) no paths between a & b use e

2) all paths use e

CONTRADICTION

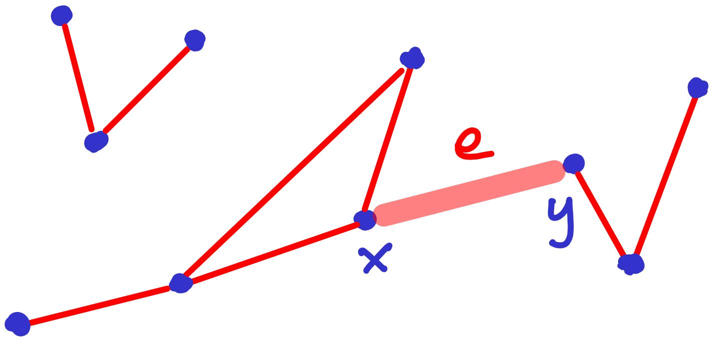
If path P uses e & path Q doesn't then e is on a cycle



but cut edges don't exist on cycles

Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Types of vertex pairs (a, b) in G:



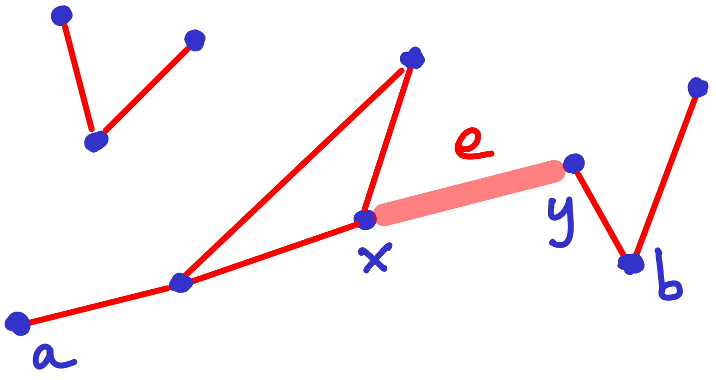
0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e

Type 0 or 1: not affected by removal of e

Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

Types of vertex pairs (a, b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e



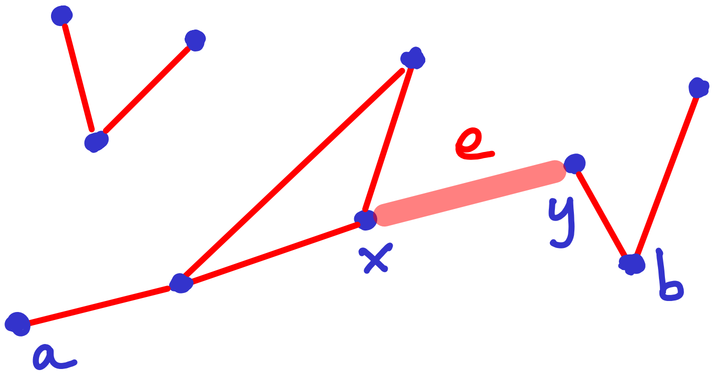
Type 0 or 1: not affected by removal of e

Type 2: implies a connects to x but not to y & b connects to y but not to x $\Big\}$ in $G - e$

Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

Types of vertex pairs (a,b) in G:

0) no path exists between a & b
1) no paths between a & b use e
2) all paths use e



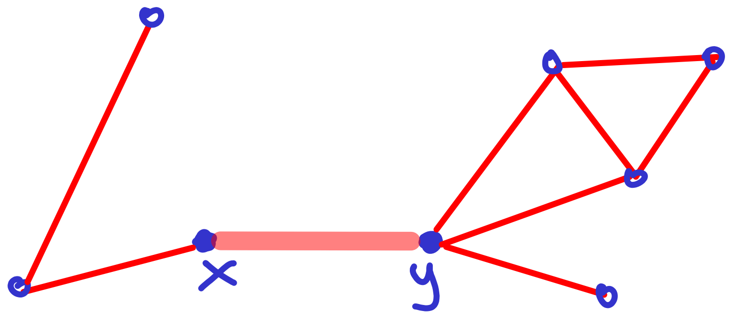
Type 0 or 1: not affected by removal of e

Type 2: implies a connects to x but not to y & b connects to y but not to x $\Big\}$ in $G-e$

↳ Type 2 partitions one component into two.

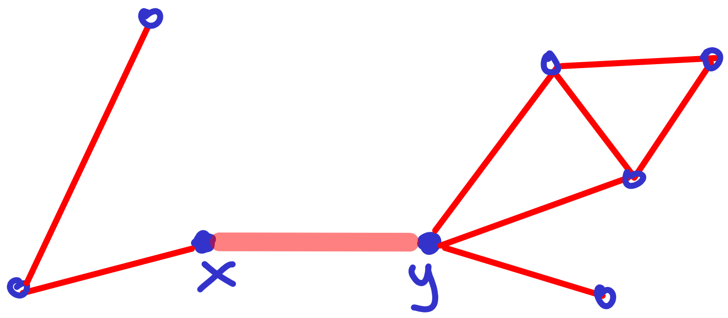
Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

* e can only affect the component it's in. So focus on connected graphs.



Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

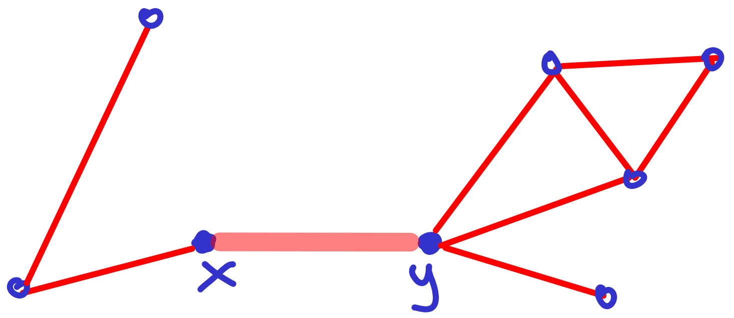
* e can only affect the component it's in. So focus on connected graphs.



Proof by contradiction.

Suppose $G - e$ has ≥ 3 components.

Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.



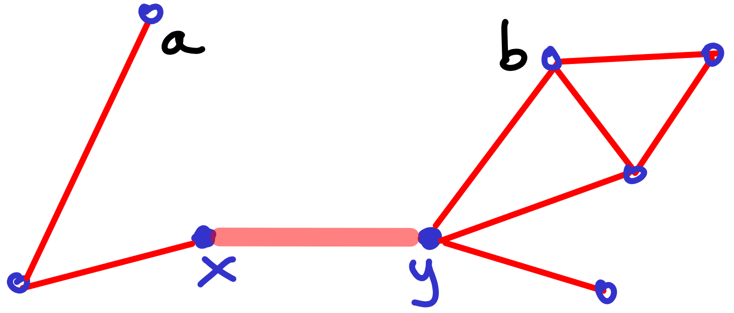
* e can only affect the component it's in.
So focus on connected graphs.

Proof by contradiction.

Suppose $G-e$ has ≥ 3 components. $\exists\, a,b,c$ in different components.

Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.

* e can only affect the component it's in. So focus on connected graphs.



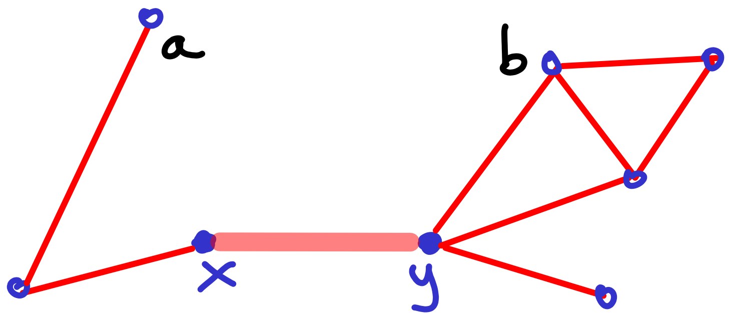
Proof by contradiction.

Suppose $G - e$ has ≥ 3 components. $\exists\, a, b, c$ in different components.

In G, all paths $a \to b$ use e } wlog $a \to x \to y \to b$

Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

* e can only affect the component it's in.
  So focus on connected graphs.



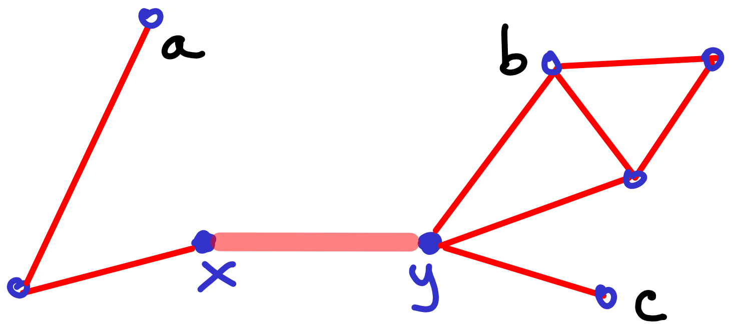
Proof by contradiction.

Suppose $G-e$ has ≥ 3 components. $\exists\, a,b,c$ in different components.

In G, all paths $a \to b$ use e } wlog $a \to x \to y \to b$

all paths $a \to c$ use e

Claim: Removing a cut edge $e = (x, y)$ increases the number of components by 1.



* e can only affect the component it's in.
So focus on connected graphs.

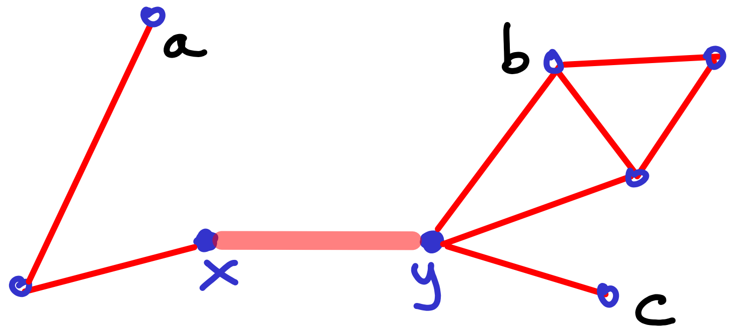
Proof by contradiction.

Suppose $G - e$ has ≥ 3 components. $\exists\, a, b, c$ in different components.

In G, all paths $a \to b$ use e } wlog $a \to x \to y \to b$

all paths $a \to c$ use e { if $a \to x \to y \to c$

Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.



* e can only affect the component it's in.
So focus on connected graphs.

Proof by contradiction.

Suppose $G-e$ has ≥ 3 components. $\exists\, a,b,c$ in different components.

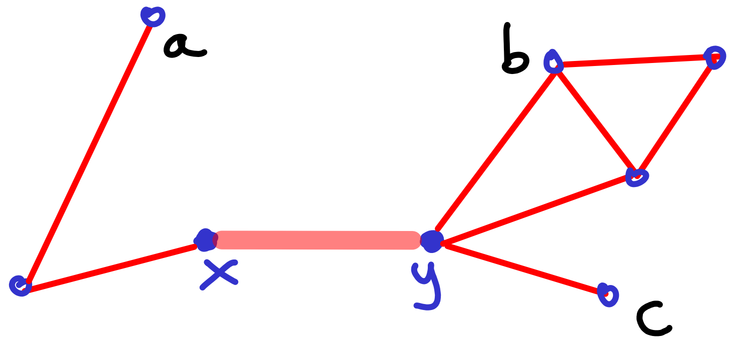
In G, all paths $a \to b$ use e } wlog $a \to x \to y \to b$

all paths $a \to c$ use e { if $a \to x \to y \to c$

$\Rightarrow$ b & c in same component

Claim: Removing a cut edge $e=(x,y)$ increases the number of components by 1.

* e can only affect the component it's in. So focus on connected graphs.



Proof by contradiction.

Suppose $G-e$ has ≥ 3 components. $\exists\, a,b,c$ in different components.

In G, all paths $a \to b$ use e } wlog $a \to x \to y \to b$

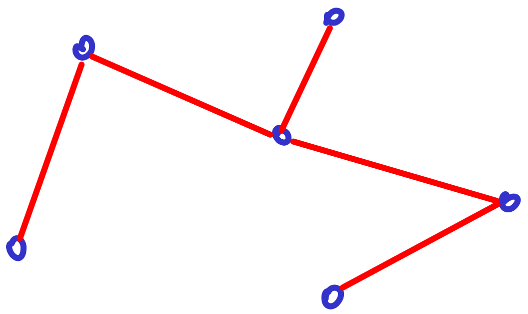
all paths $a \to c$ use e {
if $a \to x \to y \to c$ $\Rightarrow$ b & c in same component
if $a \to y \to x \to c$ $\Rightarrow$ e not a cut edge

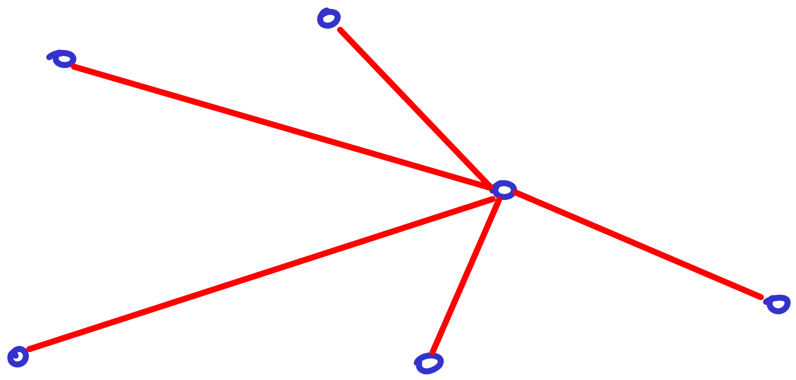
Recap

1) A cut edge can't be on a cycle.

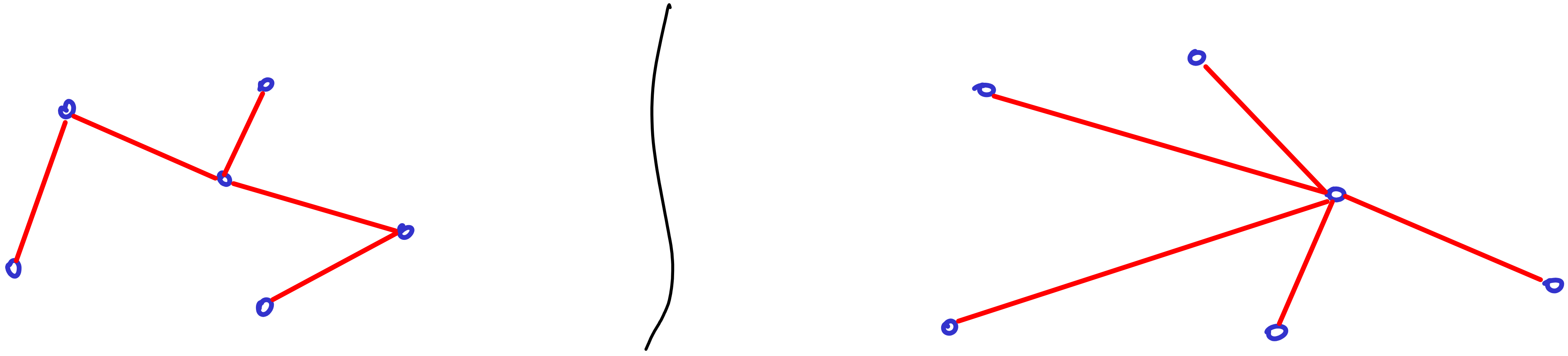
2) Removing a cut edge $e = (x, y)$ increases the number of components by 1.

TREES : CONNECTED ACYCLIC GRAPHS





TREES : CONNECTED ACYCLIC GRAPHS



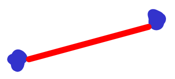
FORESTS : ACYCLIC GRAPHS (collections of trees)

$V = 1$

•

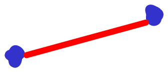
$V = 1$

$V = 2$

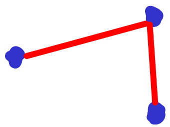


$V = 1$

$V = 2$



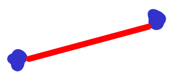
$V = 3$



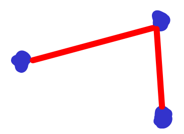
(3 isomorphs)

$V = 1$

$V = 2$

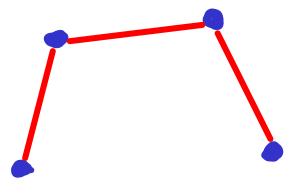


$V = 3$

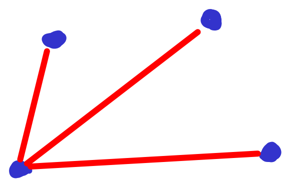


(3 isomorphs)

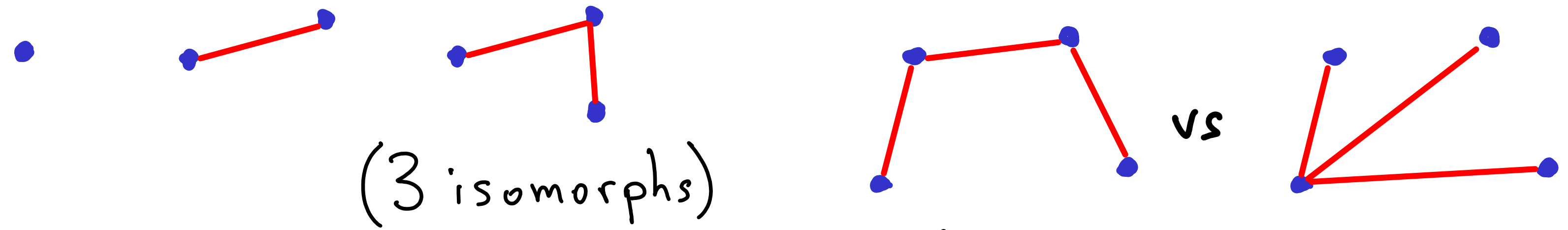
$V = 4$



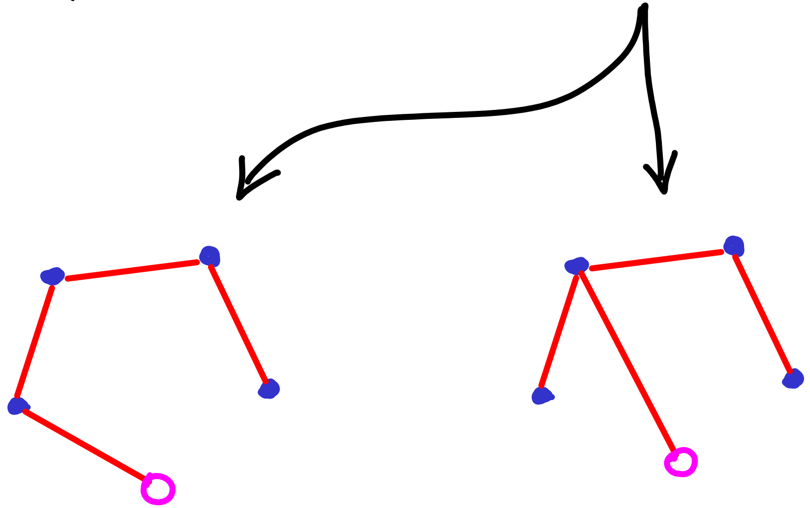
vs

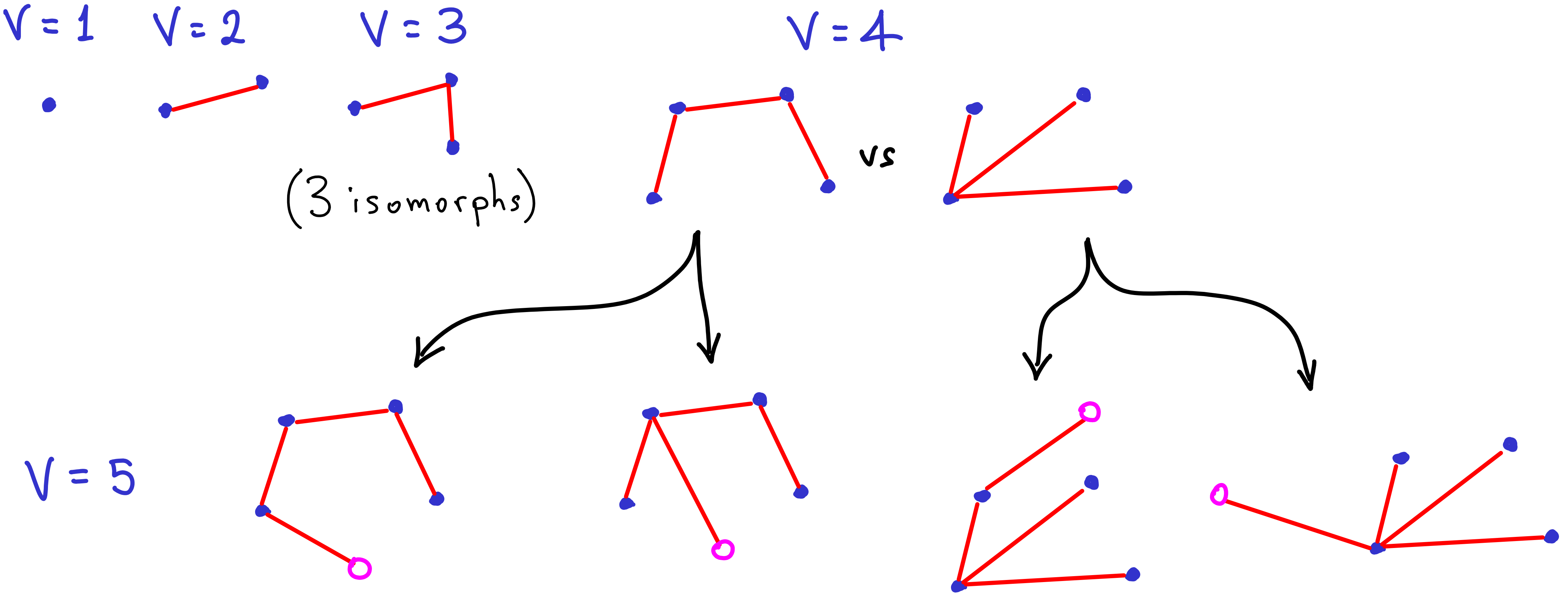


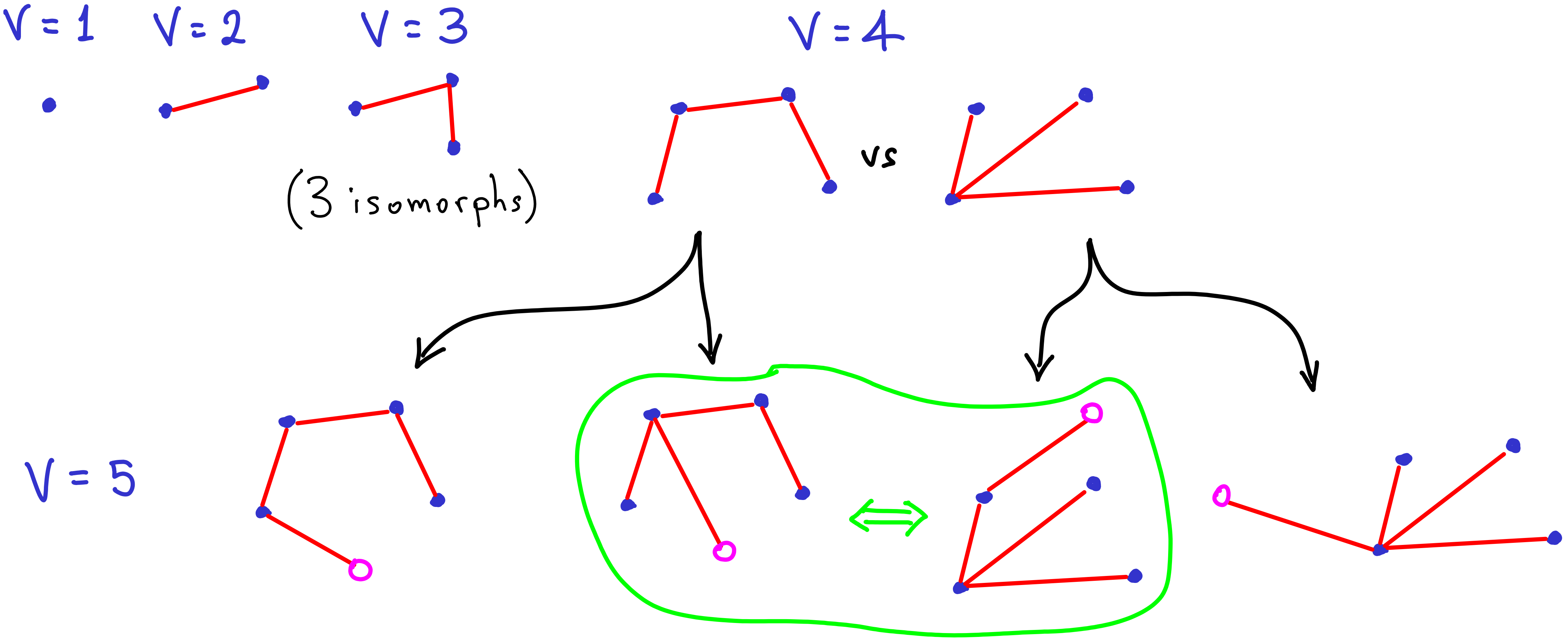
V = 1 V = 2 V = 3 V = 4



V = 5




V = 1
V = 2
V = 3
(3 isomorphs)
V = 4
vs
V = 5


V = 1
V = 2
V = 3
(3 isomorphs)
V = 4
vs
V = 5

tree $\iff$ there is a unique path between every pair of vertices

tree $\Longleftrightarrow$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a, b : a path exists (trees are connected)

next ?

tree $\Longleftrightarrow$ there is a unique path between every pair of vertices

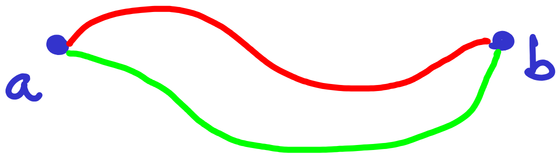
$\Rightarrow$
- for any vertices a, b : a path exists (trees are connected)
- suppose $\geqslant 2$ paths.

tree $\iff$ there is a unique path between every pair of vertices

---

$\Rightarrow$ • for any vertices a, b : a path exists (trees are connected)

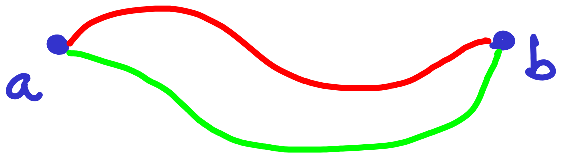
• suppose ≥ 2 paths.

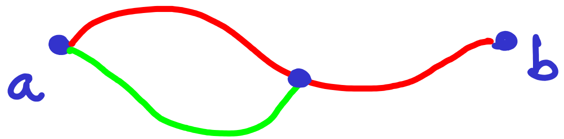


tree $\iff$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a,b : a path exists (trees are connected)

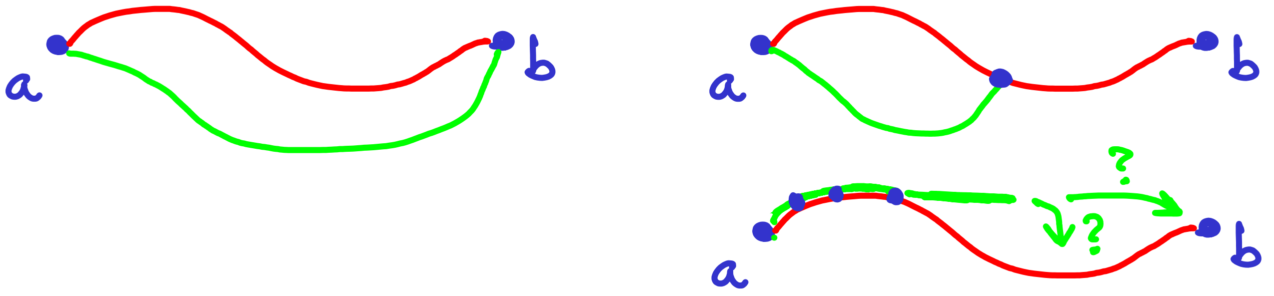
• suppose ≥ 2 paths.





tree $\iff$ there is a unique path between every pair of vertices

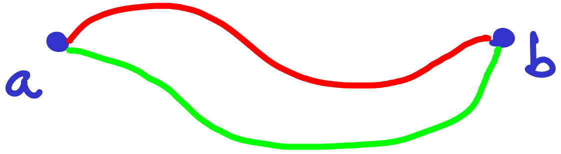
$\Rightarrow$
- for any vertices a, b : a path exists (trees are connected)
- suppose ≥ 2 paths.

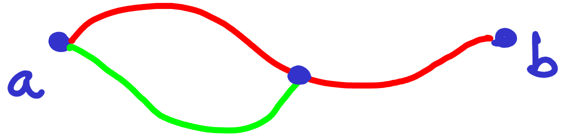


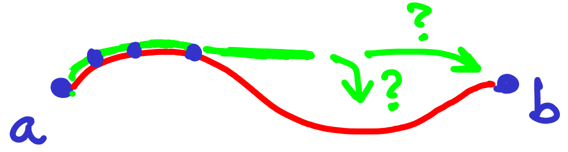
tree $\iff$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a, b : a path exists (trees are connected)

• suppose ≥ 2 paths.





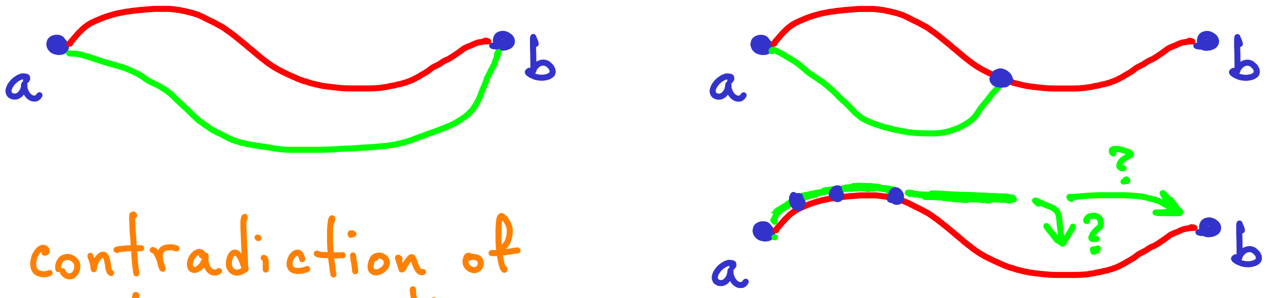


$\hookrightarrow$ cycle : contradiction of tree: acyclic

tree $\Longleftrightarrow$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a, b : a path exists (trees are connected)

• suppose ≥ 2 paths.



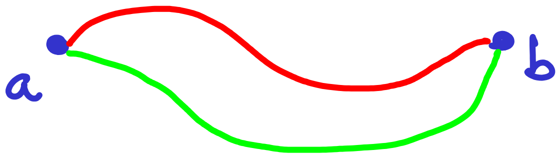
cycle : contradiction of tree: acyclic

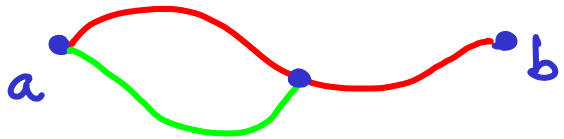
$\Leftarrow$ What 2 properties do we need to prove?

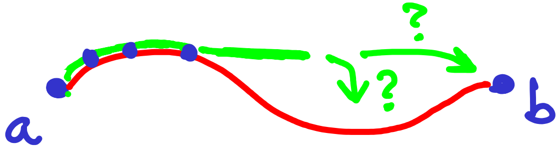
tree $\iff$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a,b : a path exists (trees are connected)

• suppose ≥ 2 paths.







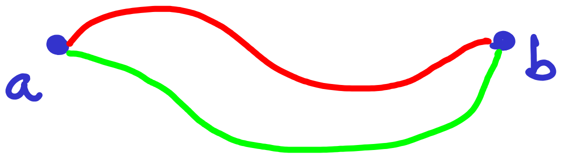
cycle : contradiction of tree: acyclic

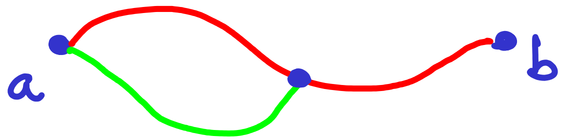
$\Leftarrow$ • if for every 2 vertices a path exists, then graph is connected

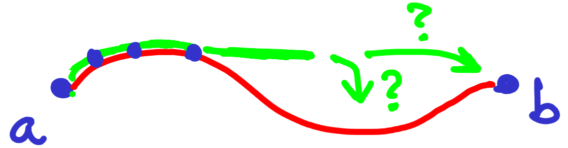
tree $\iff$ there is a unique path between every pair of vertices

$\Rightarrow$ • for any vertices a,b : a path exists (trees are connected)

• suppose ≥ 2 paths.







↳ cycle : contradiction of tree: acyclic

$\Leftarrow$ • if for every 2 vertices a path exists, then graph is connected

• if any 2 vertices are on a cycle, then they are on ≥ 2 paths but we assume unique paths, so no 2 vertices are on a cycle.

↳ acyclic

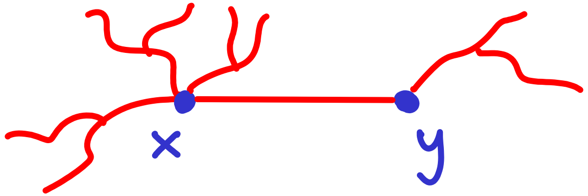
$\square$

For any connected graph,

tree $\iff$ every edge is a cut edge

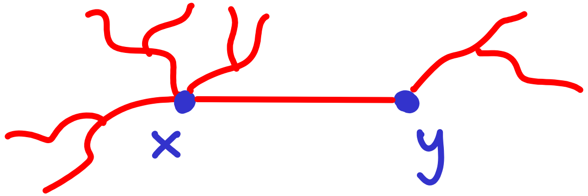
For any connected graph,

tree $\Longleftrightarrow$ every edge is a cut edge

$\Rightarrow$  $\Big\}$ for any x, y $\Big\}$ tree $\Rightarrow$ unique path from x to y

For any connected graph,

tree $\Longleftrightarrow$ every edge is a cut edge

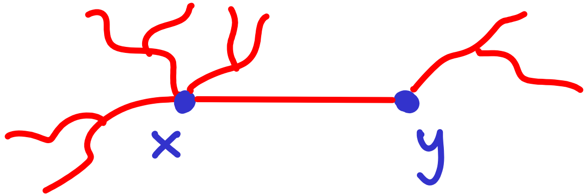
$\Rightarrow$ 

$\Big\}$ for any x, y $\Big\}$ tree $\Rightarrow$ unique path from x to y

$\Rightarrow$ $\overline{xy}$: cut edge

For any connected graph,

tree $\Longleftrightarrow$ every edge is a cut edge

$\Rightarrow$ 

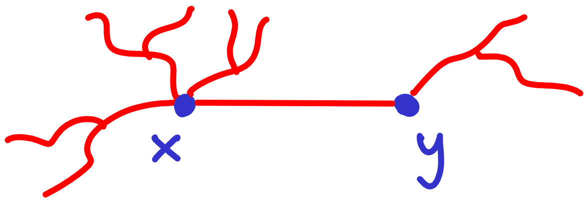
$\Bigg\}$ for any x, y $\Bigg\}$ tree $\Rightarrow$ unique path from x to y

$\Rightarrow$ $\overline{xy}$: cut edge

$\Leftarrow$ suppose graph $\neq$ tree. then...?

For any connected graph,

tree $\Longleftrightarrow$ every edge is a cut edge

$\Rightarrow$  for any x, y tree $\Rightarrow$ unique path from x to y

$\Rightarrow$ $\overline{xy}$: cut edge

$\Leftarrow$ suppose graph $\neq$ tree.

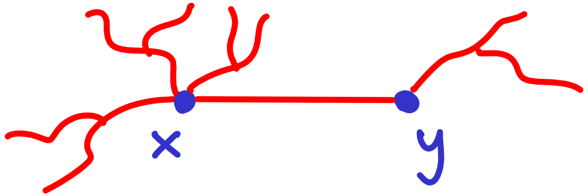
Then it has a cycle.

...?

For any connected graph,

tree $\iff$ every edge is a cut edge

$\Rightarrow$



$\left.\right\}$ for any x, y $\left.\right\}$ tree $\Rightarrow$ unique path from x to y

$\Rightarrow$ $\overline{xy}$: cut edge

$\Leftarrow$ suppose graph $\neq$ tree.

Then it has a cycle.

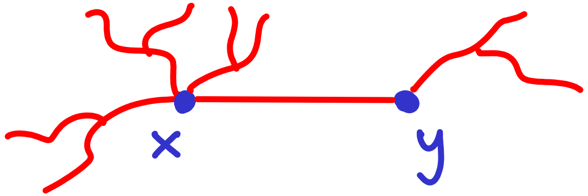
We have proved: A cut edge can't be on a cycle.

so?

For any connected graph,

tree $\Longleftrightarrow$ every edge is a cut edge

$\Rightarrow$



$\left.\right\}$ for any x, y $\left.\right\}$ tree $\Rightarrow$ unique path from x to y

$\Rightarrow$ $\overline{xy}$: cut edge

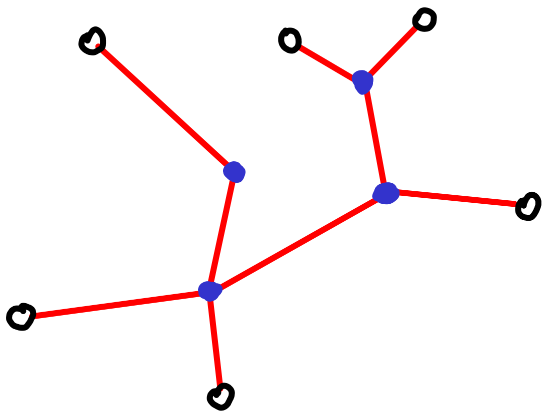
$\Leftarrow$ suppose graph $\neq$ tree.

Then it has a cycle.

We have proved: A cut edge can't be on a cycle.

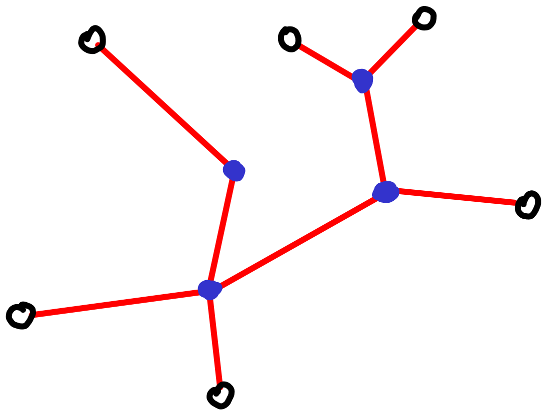
not every edge is a cut edge. (CONTRADICTION)

LEAVES : vertices of degree 1



LEAVES : vertices of degree 1

If $V \geqslant 2$, then T has $\geqslant 2$ leaves

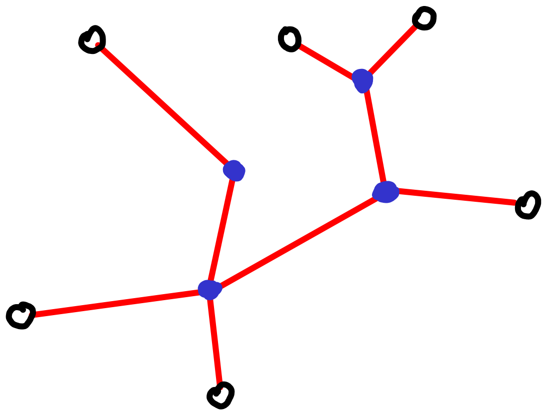


LEAVES : vertices of degree 1

If $V \geqslant 2$, then T has $\geqslant 2$ leaves

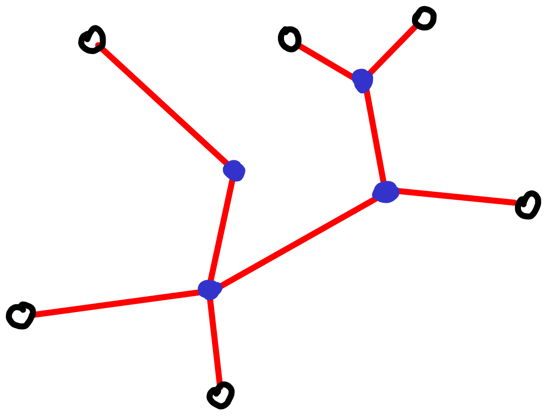
Consider longest path in T. $v_1 \dots v_k$

$(k \geqslant 2)$



LEAVES : vertices of degree 1

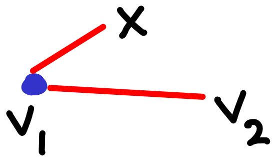
If $V \geqslant 2$, then T has $\geqslant 2$ leaves



Consider longest path in T. $v_1 \dots v_k$

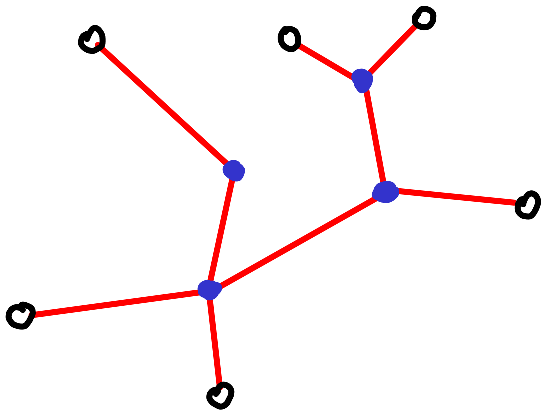
$(k \geqslant 2)$

If $v_1 \neq$ leaf, then



LEAVES : vertices of degree 1

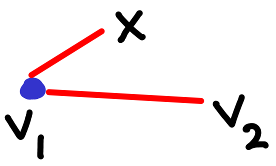
If $V \geq 2$, then T has ≥ 2 leaves



Consider longest path in T. $\quad v_1 \dots v_k$

$(k \geq 2)$

If $v_1 \neq$ leaf, then

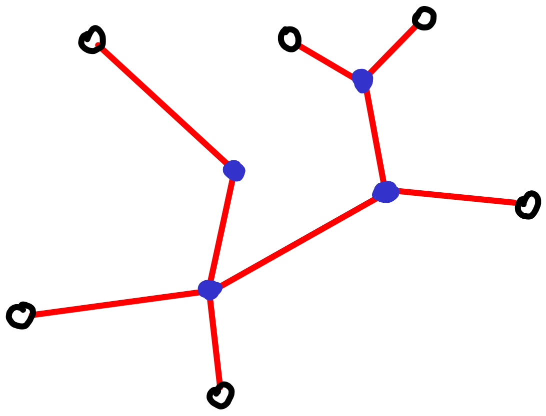


$x \neq v_i$ (not on path)

why?

LEAVES : vertices of degree 1

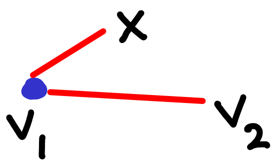
If $V \geq 2$, then T has ≥ 2 leaves



Consider longest path in T. $v_1 \ldots v_k$

$(k \geq 2)$

If $v_1 \neq$ leaf, then

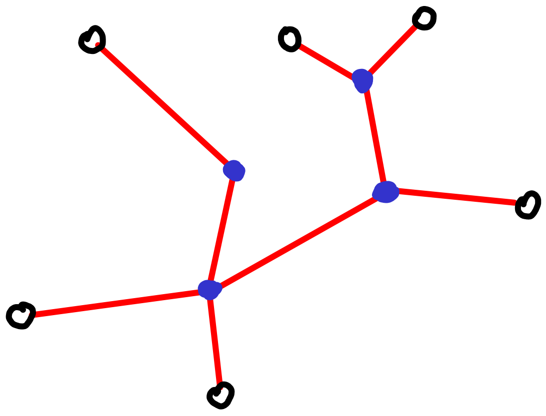


$x \neq v_i$ (not on path)

($x = v_i$ would create cycle)

LEAVES : vertices of degree 1

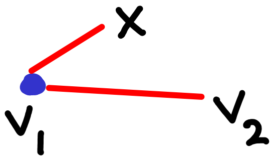
If $V \geqslant 2$, then T has $\geqslant 2$ leaves



Consider longest path in T. $v_1 \dots v_k$

($k \geqslant 2$)

If $v_1 \neq$ leaf, then



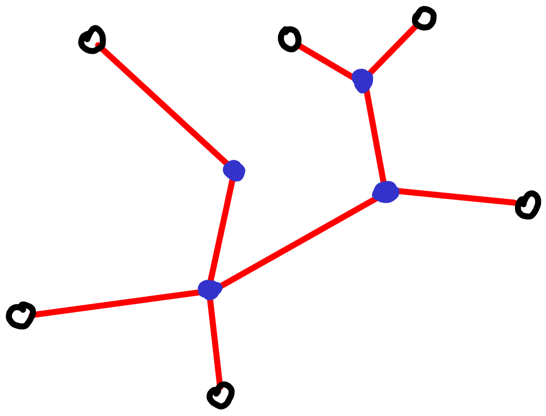
$x \neq v_i$ (not on path)

($x = v_i$ would create cycle)

Then $x v_1 \dots v_k$: longer path

LEAVES : vertices of degree 1

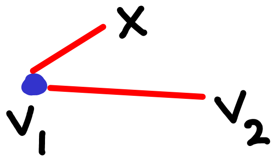
If $V \geq 2$, then T has ≥ 2 leaves



Consider longest path in T. $v_1 \dots v_k$

$(k \geq 2)$

If $v_1 \neq$ leaf, then



$x \neq v_i$ (not on path)

($x = v_i$ would create cycle)

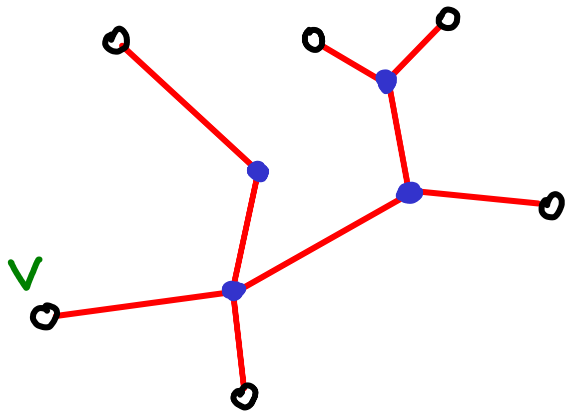
Then $x v_1 \dots v_k$: longer path : CONTRADICTION

So v_1 & v_k : leaves

If v is a leaf in tree T, then $T-v$ is a tree

If v is a leaf in tree T, then T-v is a tree

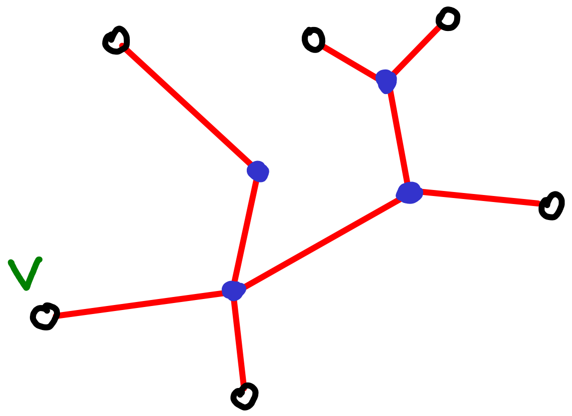
- removing v doesn't create cycles.



If v is a leaf in tree T, then $T-v$ is a tree

- removing v doesn't create cycles.

- removing v doesn't disconnect.
  ($v \neq$ cut vertex ; $T-v$ is connected)

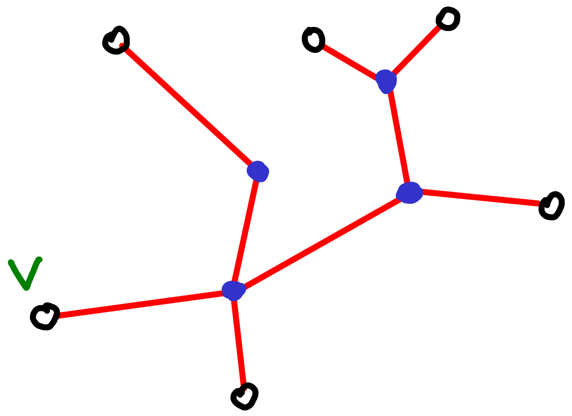


If v is a leaf in tree T, then $T-v$ is a tree

- removing v doesn't create cycles.

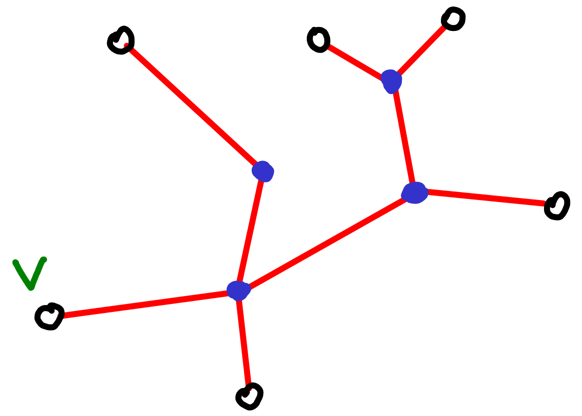
- removing v doesn't disconnect.
  ($v \neq$ cut vertex ; $T-v$ is connected)

↳ if v were a cut vertex, then $\exists\, a,b$ ($a \neq v$, $b \neq v$) s.t. any path $a \to b$ must use v.



If v is a leaf in tree T, then T-v is a tree

- removing v doesn't create cycles.
- removing v doesn't disconnect.
  ($v \neq$ cut vertex ; T-v is connected)

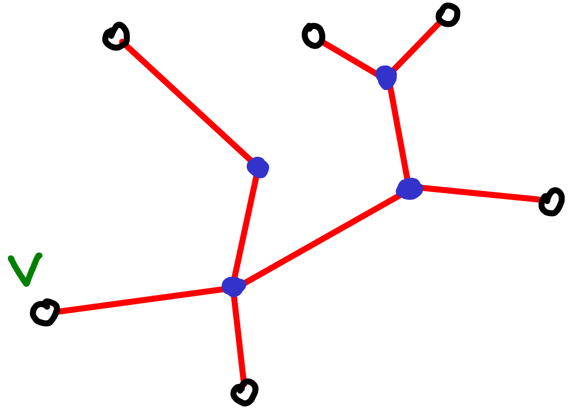


↳ if v were a cut vertex, then $\exists\, a,b$ $(a \neq v, b \neq v)$ s.t.
any path $a \to b$ must use v.

in fact,
"the only path"
(T : unique paths)

If v is a leaf in tree T, then $T-v$ is a tree

- removing v doesn't create cycles.



- removing v doesn't disconnect.
  ($v \neq$ cut vertex ; $T-v$ is connected)

↳ if v were a cut vertex, then $\exists\ a,b$ $(a \neq v,\ b \neq v)$ s.t.
any path $a \to b$ must use v.
But v is a dead end:
can't be part of such a path

in fact,
"the only path"
(T : unique paths)

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

Hypothesis: for $2 \leq k < n$, statement holds.

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

Hypothesis: for $2 \leq k < n$, statement holds.

Suppose T has n vertices. Find a leaf v & delete.

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

Hypothesis: for $2 \leq k < n$, statement holds.

Suppose T has n vertices. Find a leaf v & delete.

- v had degree 1, so we delete 1 edge.

If v is a leaf in tree T, then $T-v$ is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

Hypothesis: for $2 \leq k < n$, statement holds.

Suppose T has n vertices. Find a leaf v & delete.

- v had degree 1, so we delete 1 edge.
- $T-v$ is a tree, w/ $n-1$ vertices $\rightarrow$ $n-2$ edges.

If v is a leaf in tree T, then T-v is a tree

This allows us to use induction

ex: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

pf: Base case: $n=2$  trivial

Hypothesis: for $2 \leq k < n$, statement holds.

Suppose T has n vertices. Find a leaf v & delete.

- v had degree 1, so we delete 1 edge.
- T-v is a tree, w/ n-1 vertices $\rightarrow$ n-2 edges.
- Replace v: total edges $= n-2+1 = n-1$

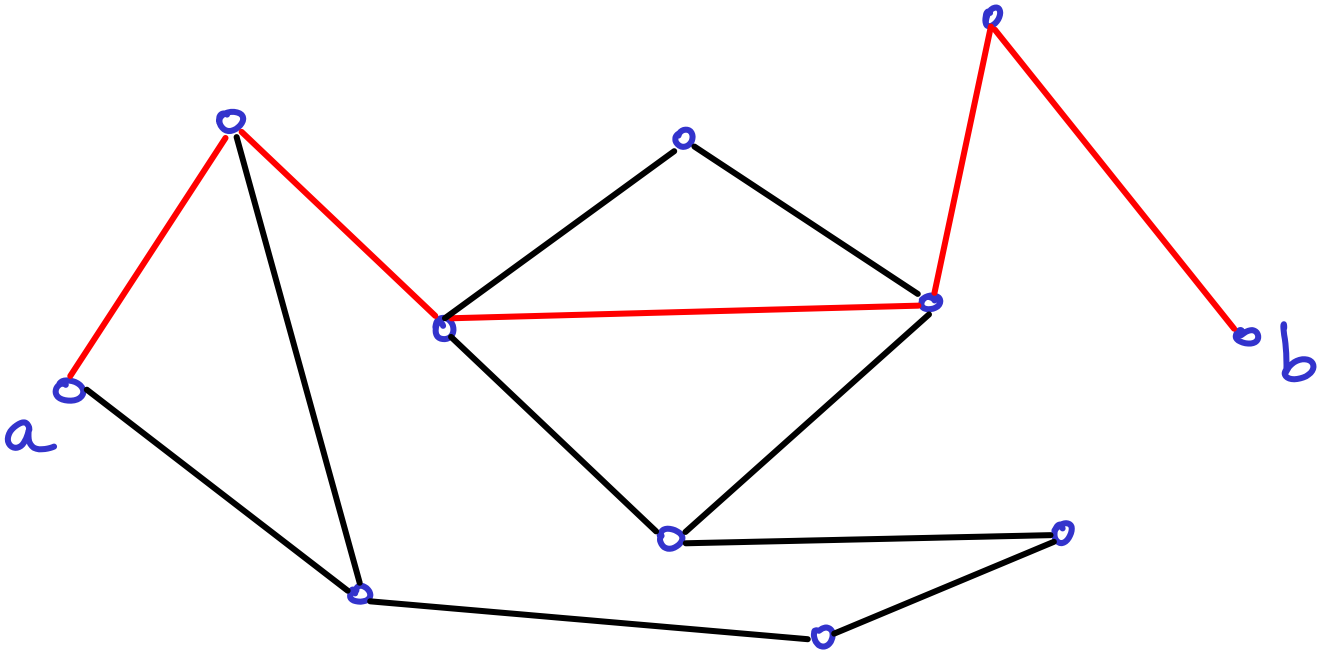
Proved: if $|V(T)| = n \geq 2$ then $|E(T)| = n-1$

Also true: for connected G with $n \geq 1$ vertices,
if $|E(G)| = n-1$ then G is a tree

See p. 354

Also defines spanning trees

DISTANCE IN GRAPHS



$d(a,b) = 5$

$d(a,b)$ = length of shortest path between a & b

Find minimum distance: covered in Algorithms course