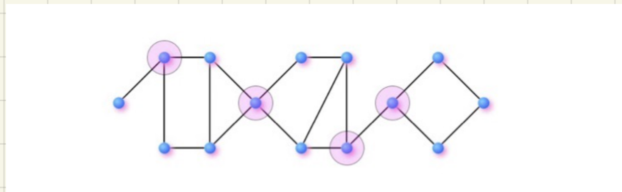
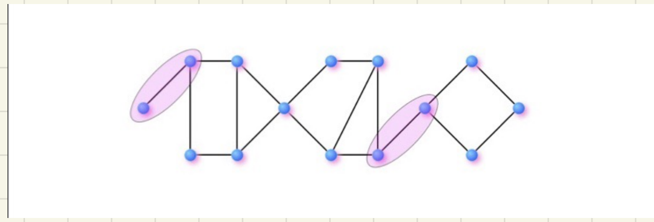
Graph Connectivity

- In an undirected graph → how connected is the graph?
  - ↳ articulation pts are vertices that disconnect graph
  - ↳ bridges are edges that do so





- Directed graph → $u \& v \in V$ strongly conn iff $\exists$ path $u \to v$ and $v \to u$
  - → graph is strongly conn iff $\forall\, u,v \in V$

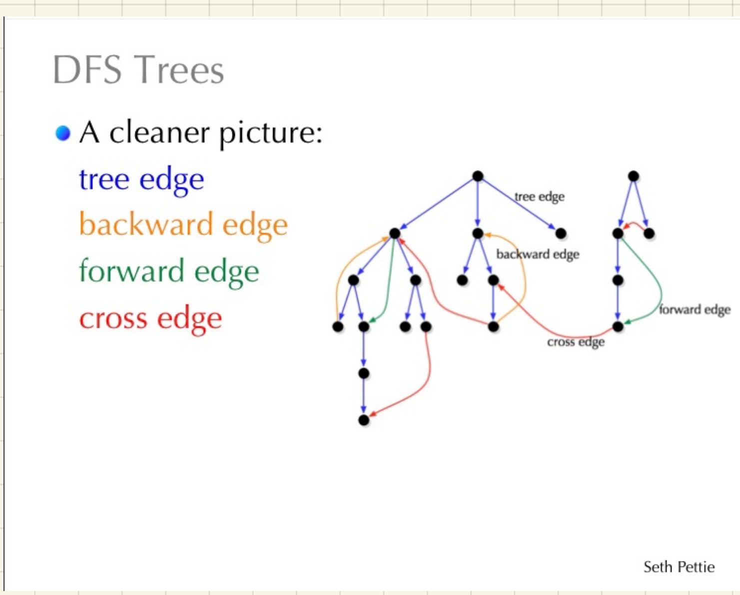
Depth First Search

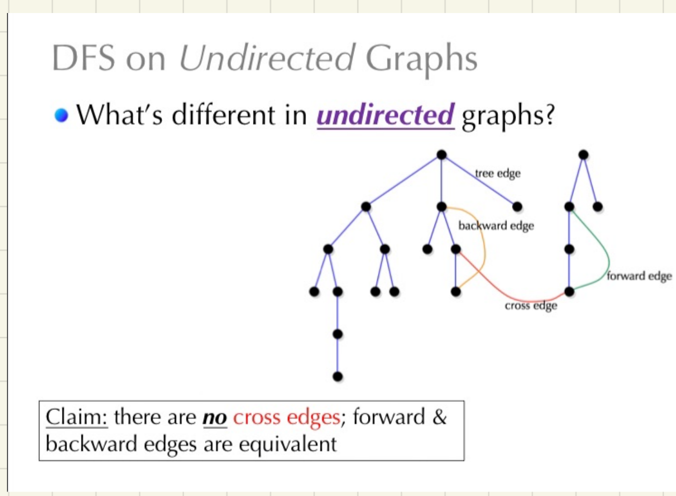
- Many connectivity problems can be solved with ***depth first search***
  - All vertices are undiscovered (white), discovered (gray), or finished (black)
- DFS-recur(u) : DFS starting from vertex u
  - Mark u *discovered*
  - For each outgoing edge (u,v)
    - If v is not already discovered/finished, call DFS-recur(v)
  - Mark u *finished*

→ some may not be reachable from u

so full Alg

DFS:
while $\exists$ undiscovered vertex u
DFS - recur(u)





If G is strongly conn $\Rightarrow \forall u$, u reaches everything

↳ reverse the edges & run DFS again

Strong Connectivity

- Given a ***directed*** graph G=(V,E), is G strongly connected?
- An algorithm:
  - Let G^T be G with all edges reversed ((u,v) becomes (v,u))
  - Pick any vertex s
  - Run DFS-recur(s) in G → s reachable from u in G
  - Run DFS-recur(s) in G^T → u is reachable from v $\forall v$
  - If both DFS-recur searches discover all vertices
    - Return 'true'
  - Otherwise return 'false'

Seth Pettie

strongly connected components

↳ set of strongly conn comp of G : $\{C_1, \ldots, C_k\}$

$C_i \subseteq V$

Strong Connectivity

- Define a relation SC ⊆ V×V
- SC(u,v) iff u and v are strongly connected

  (there are paths from u to v and v to u)
- Claim: For any directed graph, SC is an ***equivalence*** relation (reflexive, symmetric, transitive)

  Why?

Correctness pf

Lemma 1 (White path)

DFS_recur on u $\exists$ a white path from $u \to v$ then v is a descendant of u

Suppose not! let y be first non-desc of u on path

$(u) \longrightarrow \bigcirc \to \cdots (x) \to (y) \to \cdots \to (v)$

$\Rightarrow$ y desc of x $\Rightarrow$ y desc of u oops

↳ since y is white!

↳ will be reached by x.

Finding SCCs

- Definition: $G^T = (V,E')$, $(u,v) \in E'$ iff $(v,u) \in E$

  (same edges, opposite direction)
- *Note:* SCCs of G^T same as those in G
- The Algorithm:

  (1) Run DFS in G, sort vertices by ***finish time***

  (2) Run DFS in G^T
  - Whenever you choose a new root vertex u, *always choose the u maximizing f_u*

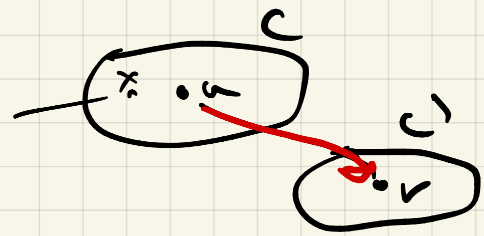
  (3) Trees in DFS forest from (2) are SCCs in G

Lemma 1 $f_u \leftarrow$ finishing time of u. If $C \subseteq V$

$$f_C = \max_{u \in C} f_u$$

Key lemma Spse strongly conn comp C, C'
& $(u,v) \in E$ with $u \in C$, $v \in C'$

$$\Rightarrow f_C > f_{C'}$$



Pf $x \rightarrow \{$first vertex disc by DFS$\}$ $\rightarrow C, C'$

Case 1| $x \in C$.
White path $\rightarrow$ C, C' are desc of x.

$$\Rightarrow f_x = f_C \geq f_e \quad \forall e \in C \cup C' \setminus \{x\}$$

$$\geq f_{C'}$$ as u finishes strictly later than v.

Case 2| $x \in C'$. Cannot be edge from $C' \rightarrow C$ otherwise, $C' \cup C'$ strongly conn comp oops.

↳ when DFS_recur(x) returns, all of C' is finished but all of C is undiscovered (white)

Proof corr| Induction on time

(2) DFS recur on roots $u_1, u_2, \dots, u_k$

Assume DFS recur(u_1) ... DFS_recr(u_i) form SSC(u_1), ..., SSC(u_i)

Claim 1 $SSC(u_{i+1}) \subseteq T_{i+1}$ $\rightarrow$ $\boxed{\text{tree}}$ of $i+1$

Pf by contra: if $y \in SSC(u_{i+1}) \setminus T_{i+1}$
can't be in $T_1, \dots, T_{i+1}$ $\Rightarrow \exists$ white path from $u_{i+1} \rightarrow y$ oops by white path lemma

Claim 2 $SSC(u_{i+1}) \supseteq T_{i+1}$

pf by contra $y \in T_{i+1} \setminus SSC(u_{i+1})$

$f_y = f_C > f_{C'} = f_{u_{i+1}}$

$\Rightarrow$ would've chosen y ins