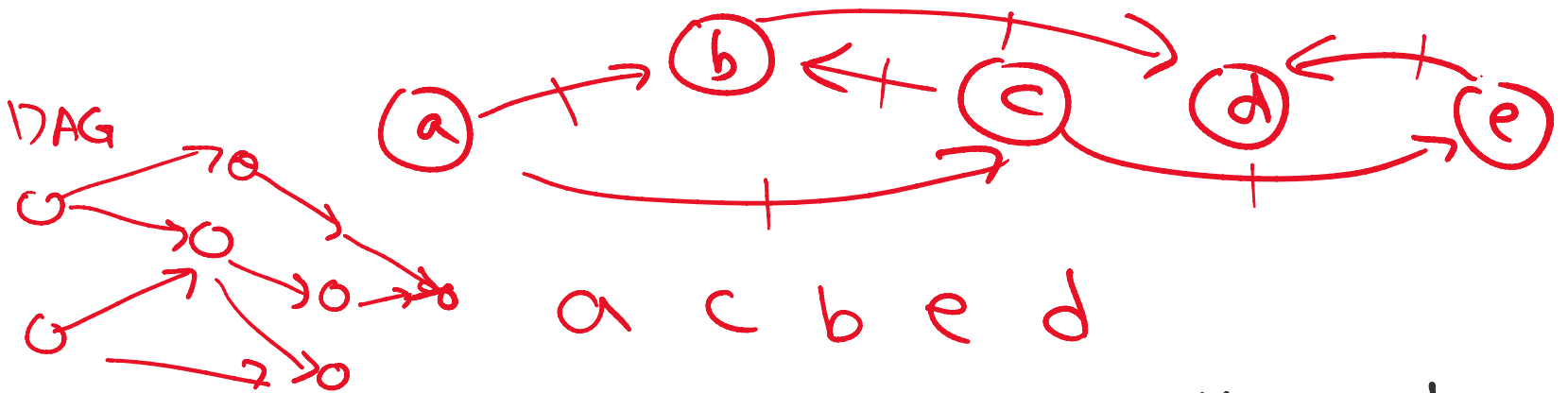


Applications of DFS

Topological Sort

Given dir graph $G=(V, E)$,
find a vertex ordering s.t.

$$\forall (u, v) \in E \Rightarrow u \text{ appears before } v$$



Rmk: if G has cycle, no sol'n exists
So assume G is acyclic (DAG)

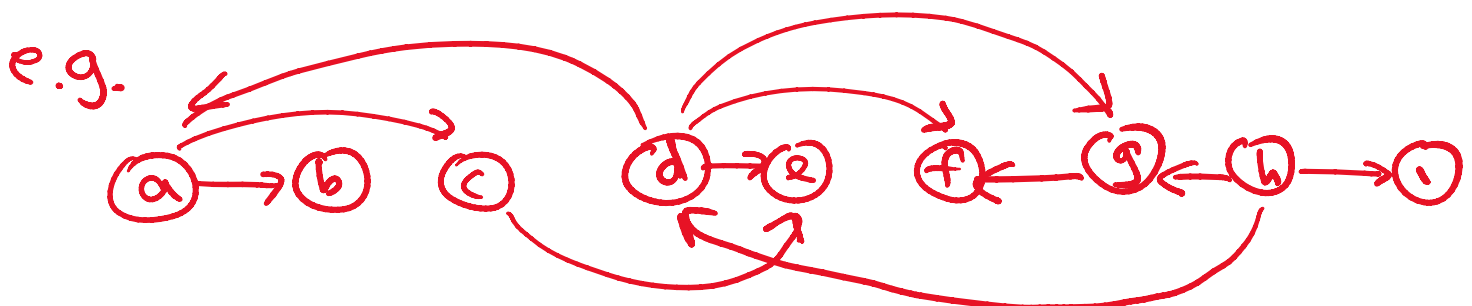
first idea - find a source vertex u
 $\hookrightarrow \text{in-deg}(u) = 0$

output u
remove u & repeat

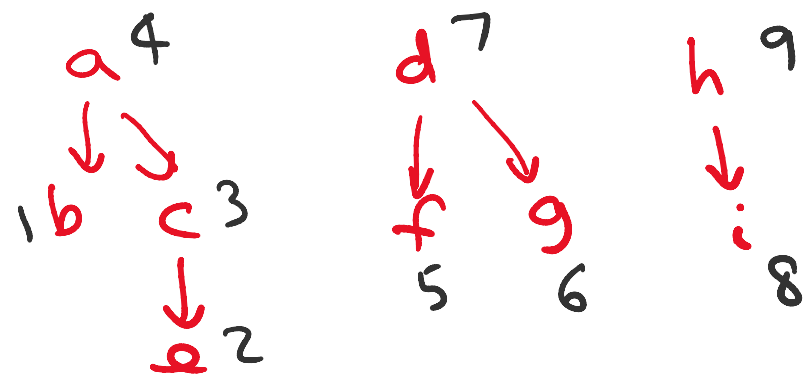
how to find a source?

run DFSAll(G)

pick $u =$ last vertex finished

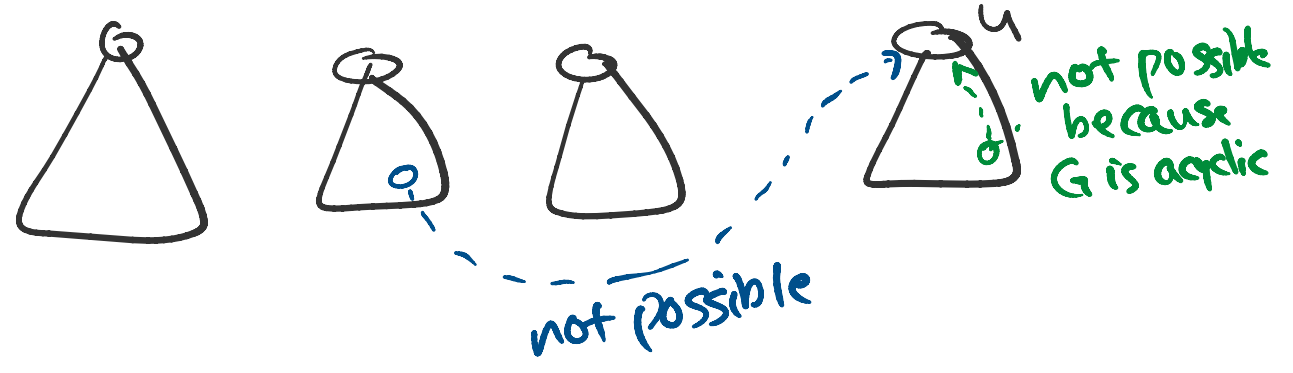


DFS trees



finish order: (postorder)
 b, e, c, a, f, g, d, i, h
 reverse finish order:
 h, i, d, g, f, a, c, e, b

Correctness Pf (Sketch): By contradiction.



How to remove? do nothing! (no need to re-run DFS)

Final Alg'm:

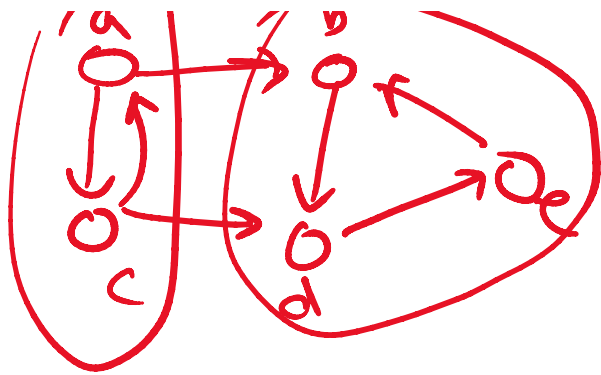
1. run DFSAll(G)
2. output vertices in reverse order of finish

⇒ $O(m+n)$ time

Corollary \exists topological sort \iff G is a DAG.

Strongly Connected Components





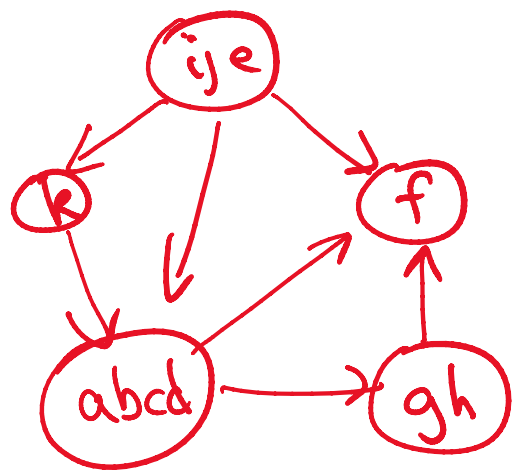
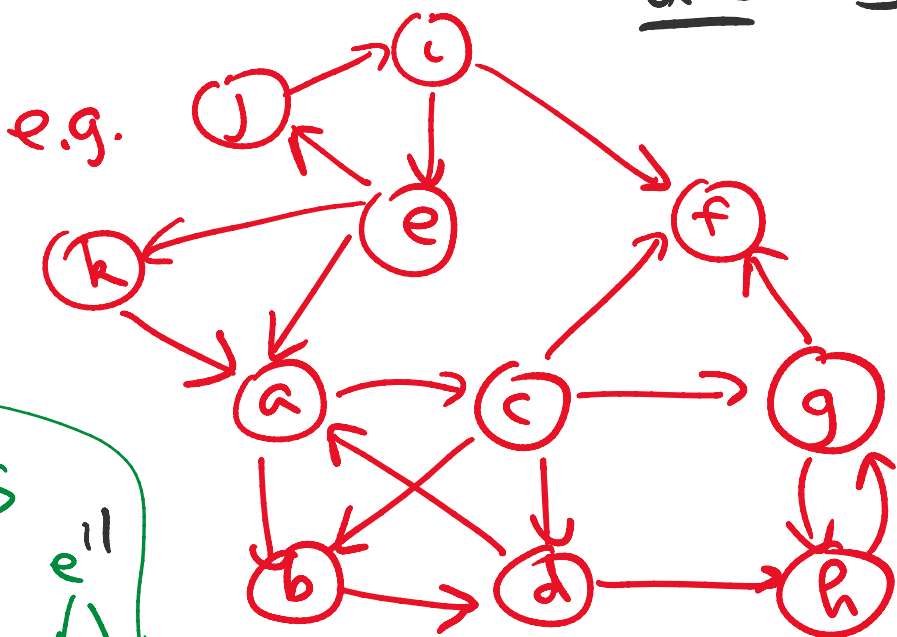
Given dir graph $G=(V,E)$,
partition V into components s.t.

u, v in same component

$\iff \exists$ path $u \rightsquigarrow v$

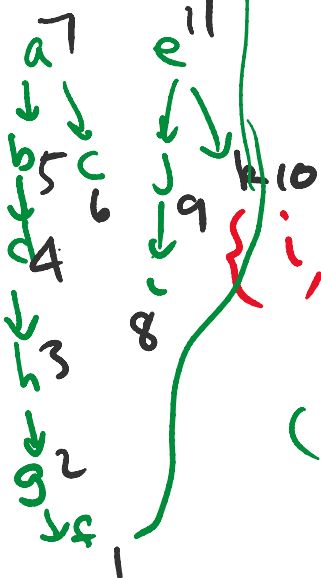
and \exists path $v \rightsquigarrow u$

(equiv relation)



Meta-graph
(must be a DAG)

DFS



$\{i, j, e\}, \{a, b, c, d\}, \{g, h\}, \{f\}, \{k\}$

(appl'n - control flow in programs, ...
Simplifying dir graphs into DAG)

naive approaches

- test reachability for every pair
by $O(n^2)$ BFS/DFS $\Rightarrow O(n(m+n))$ time
- find cycles
contract, repeat

History: Purdom '68 $O(n^2)$
" " '71 $O(m+n \log n)$

History: Purdom '68 $O(n)$
 Munro '71 $O(m+n \log n)$
 Tarjan '72 $O(m+n)$ complicated
 ⇒ Kosaraju '78 } $O(m+n)$ simpler
 Sharir '81 }

first idea - find a vertex u in some source component of meta-graph

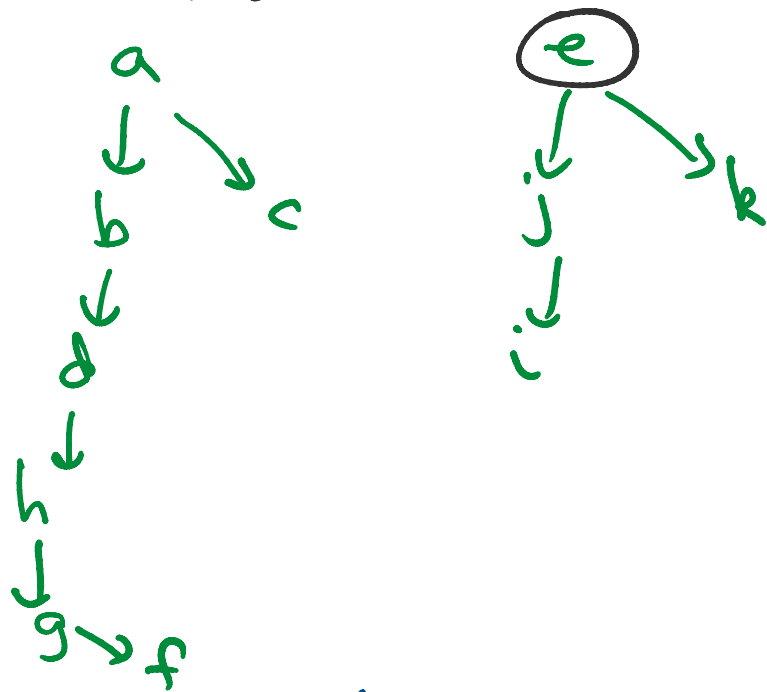
find u 's component
 remove & repeat

how to find in a vertex in a source component?

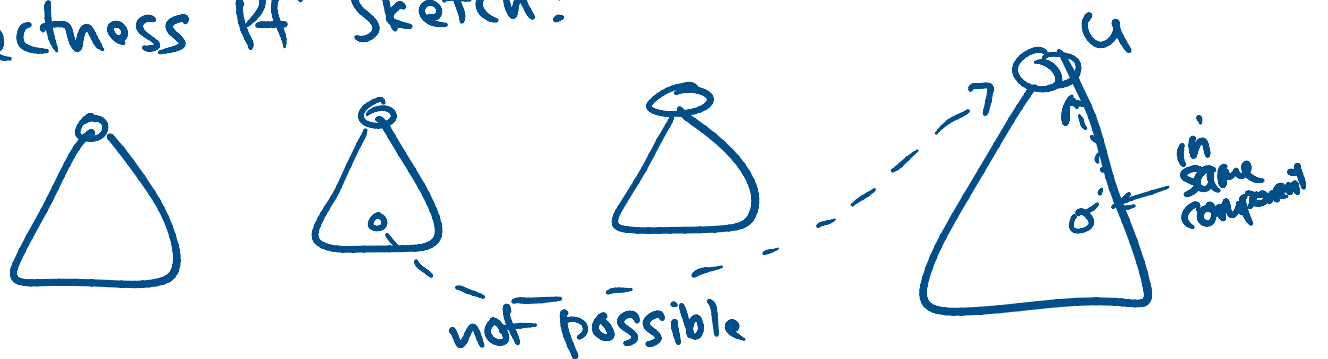
run DFSAll(G)

Pick u = last vertex finished

e.g. DFS trees



Correctness PF Sketch:



how to find u 's component?

all vertices v s.t. $v \rightsquigarrow u$

by running DFS in reverse graph G^r from u

how to remove?

do nothing

(no need to re-run the 2 DFSs)

Final Algm:

1. run DFSAll(G), label vertices by finish order

2. run DFSAll(G^r), preferring larger labels when picking roots

Simple!

\Rightarrow $O(m+n)$ time

