MA 515: Introduction to Algorithms & MA353 : Design and Analysis of Algorithms [3-0-0-6]

Lecture 14

http://www.iitg.ernet.in/psm/indexing_ma353/y09/index.html

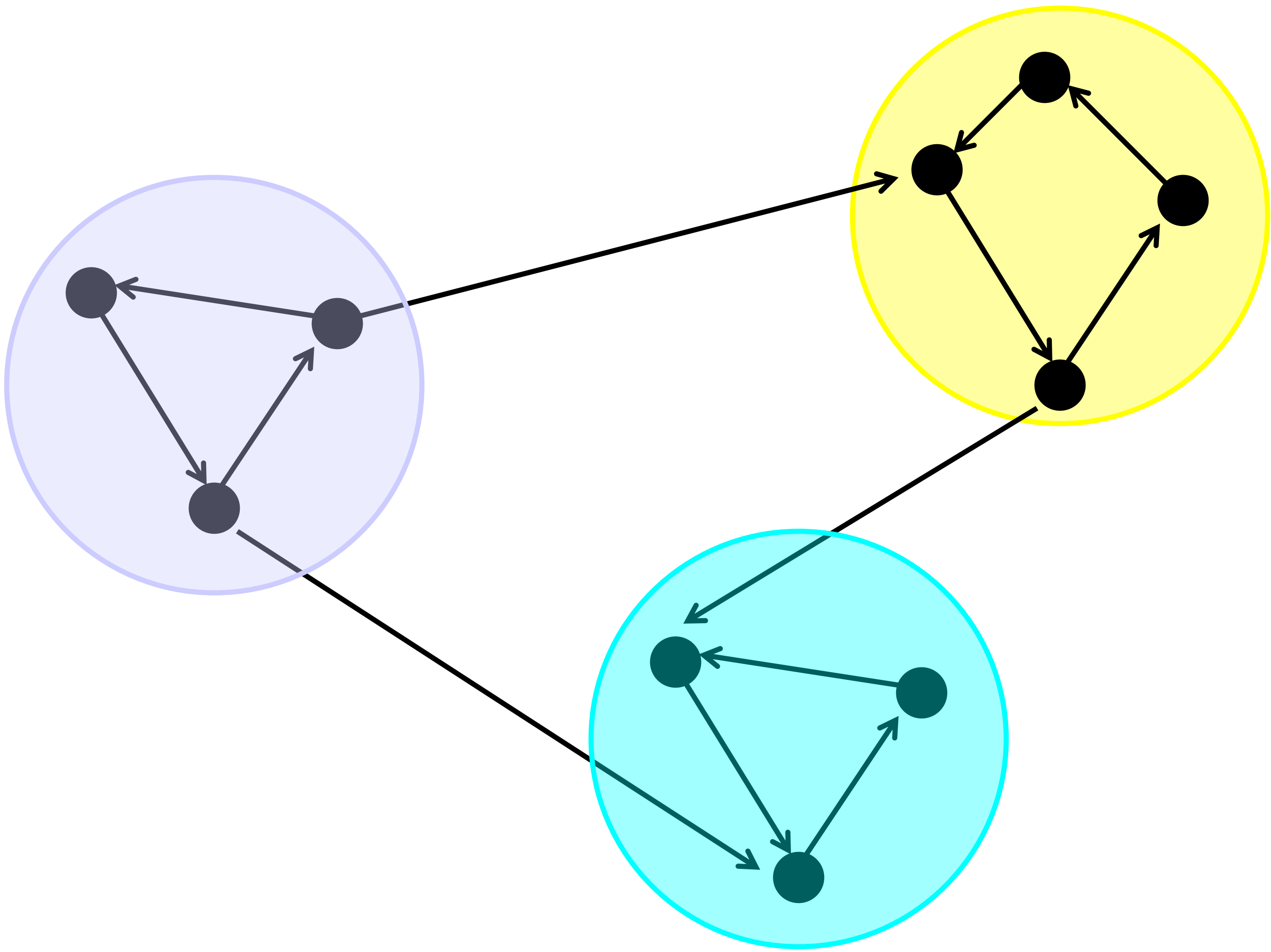
Partha Sarathi Manal

psm@iitg.ernet.in

Dept. of Mathematics, IIT Guwahati

Mon 10:00-10:55 Tue 11:00-11:55 Fri 9:00-9:55
Class Room : 2101

Example: Strongly Connected Components

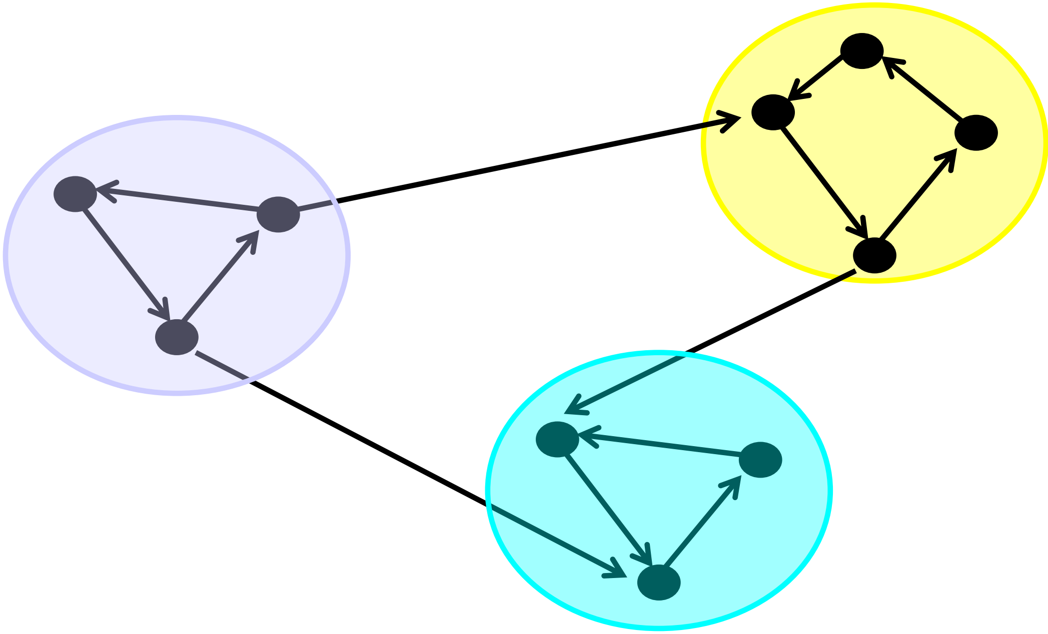


Strongly Connected Components

Def: A subset $C \subseteq V$ is a strongly connected of G if $\forall$ x, y $\in$ C ($x \rightarrow y \wedge y \rightarrow x$). A subset $C \subseteq V$ is a strongly connected component (SCC) of G if it is maximally strongly connected: no proper superset of C is strongly connected.

Strongly Connected Components

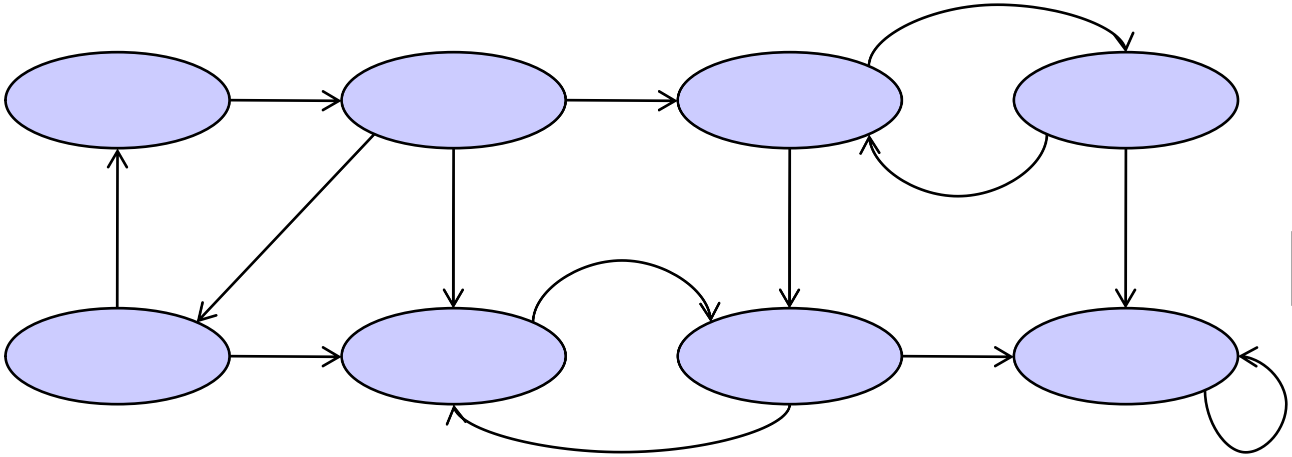
- Any two vertices in a SCC lie on a cycle.
- SCCs form a partition of the vertex set.



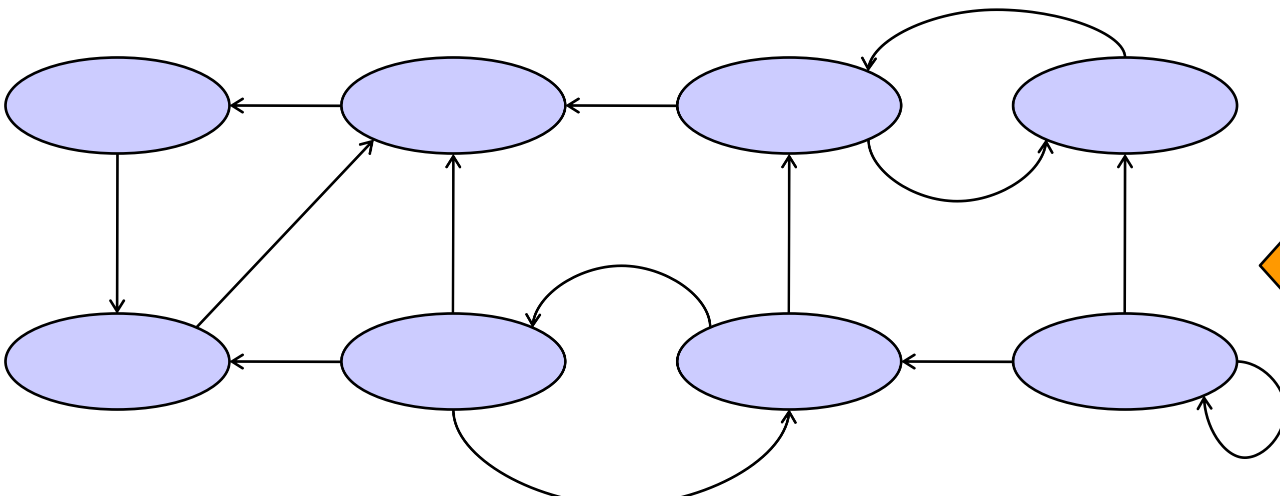
SCC algorithms

- Warshall's Algorithm: 1962
- Tarjan's Algorithm: 1972
- **Kosaraju's Algorithm: 1978**

Transpose Digraph



G=(V,E)



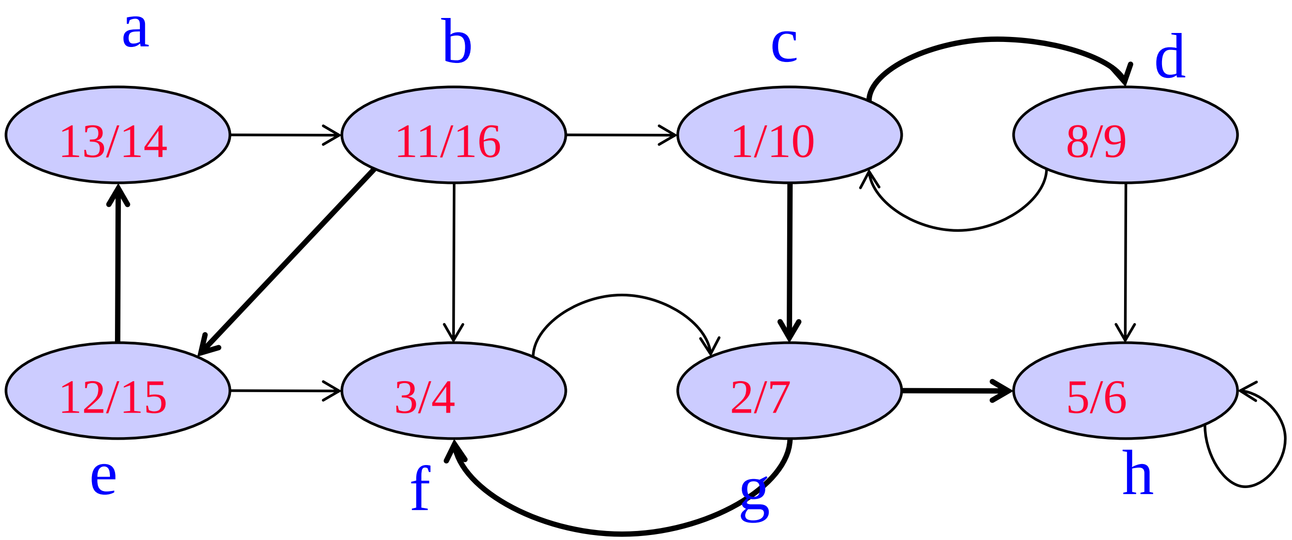
G^T=(V,E^T) where $E^T=\{(u,v): (v,u) \in E\}$

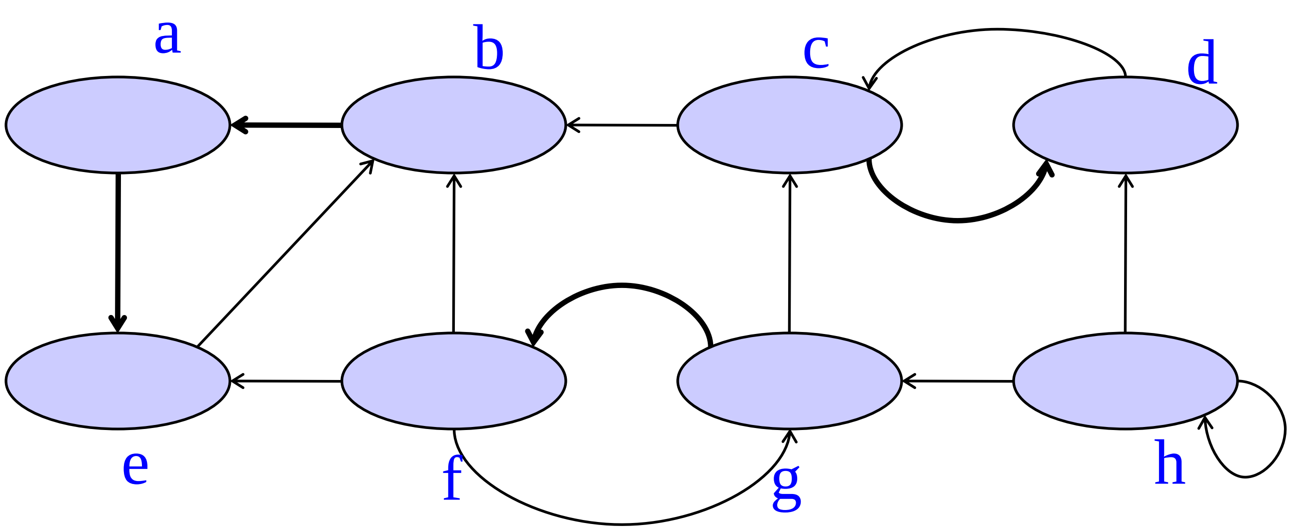
G^T can be computed from G in

O(V+E)

time

DFS on the Digraph





Kosaraju's Algorithm

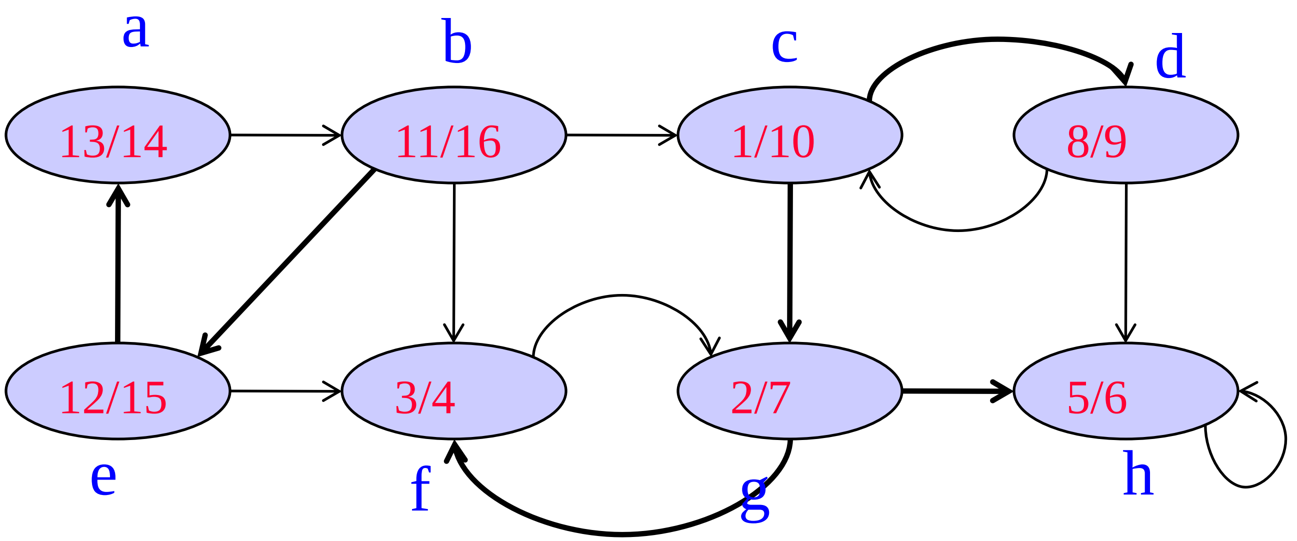
1. Perform DFS on G, store completion numbers.
2. Perform DFS on G^T, where vertices are ordered by decreasing completion numbers.
3. Each tree in the second DFS is a SCC of the graph.

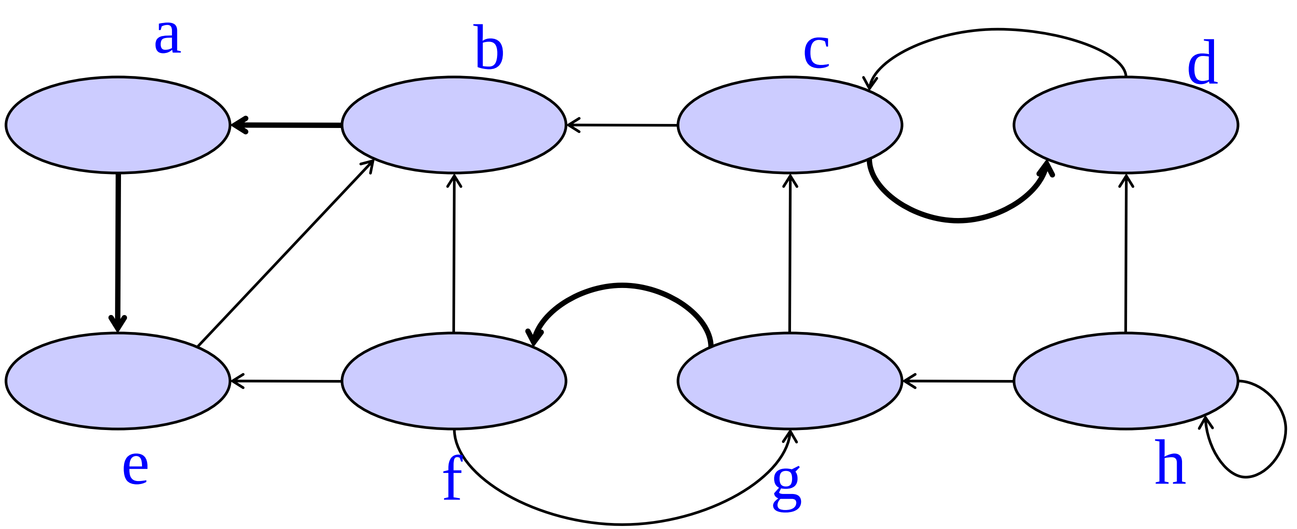
Another version of Kosaraju's Algorithm

Strongly-Connected-Components(G)

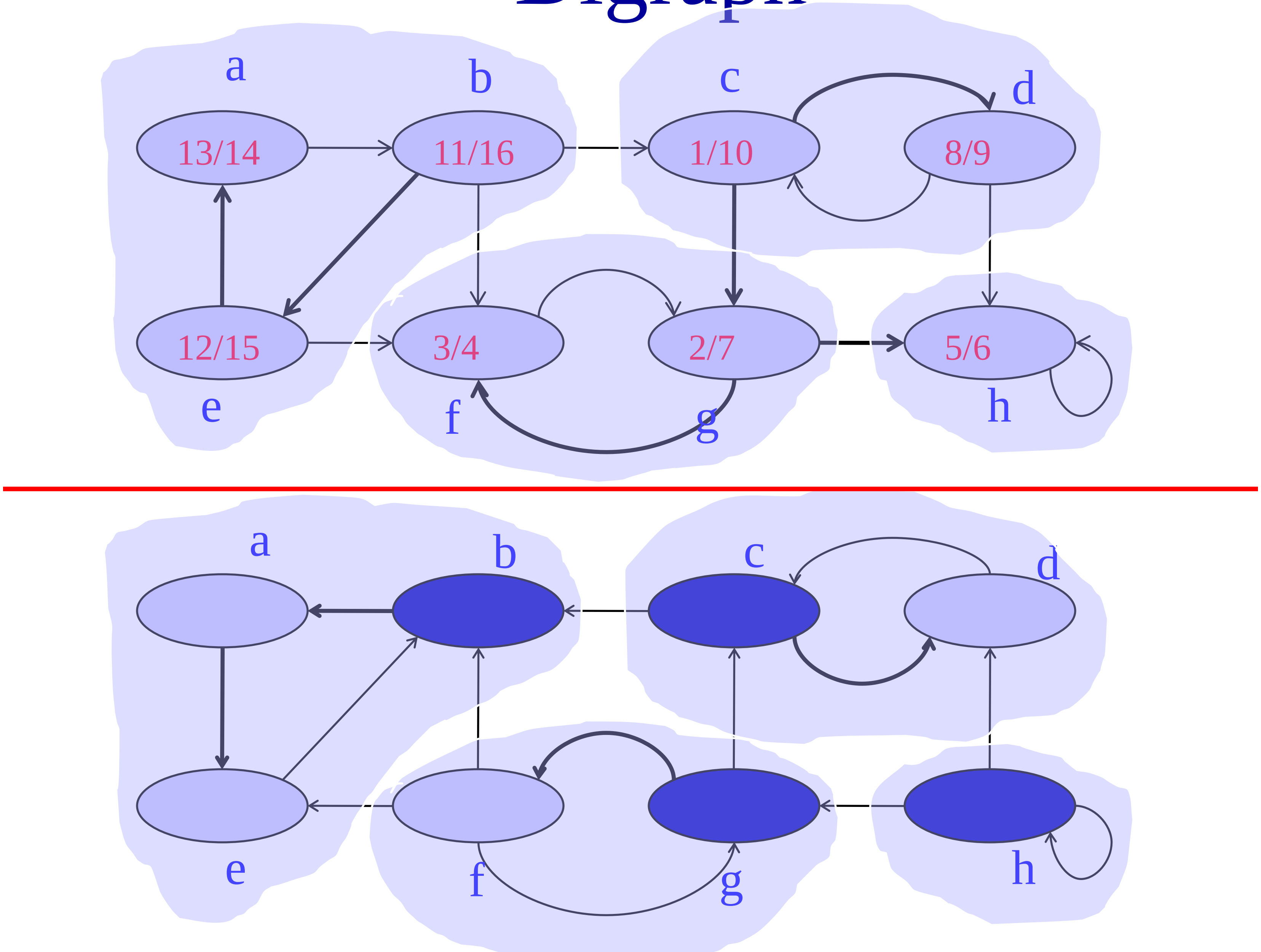
1. Call DFS to compute $f[u]$ $\forall u$.
2. Compute G^T
3. Call DFS(G^T), in order of decreasing $f[u]$.
4. Output the vertices of each tree in the depth-first Forest as a separate SCC.

The Example

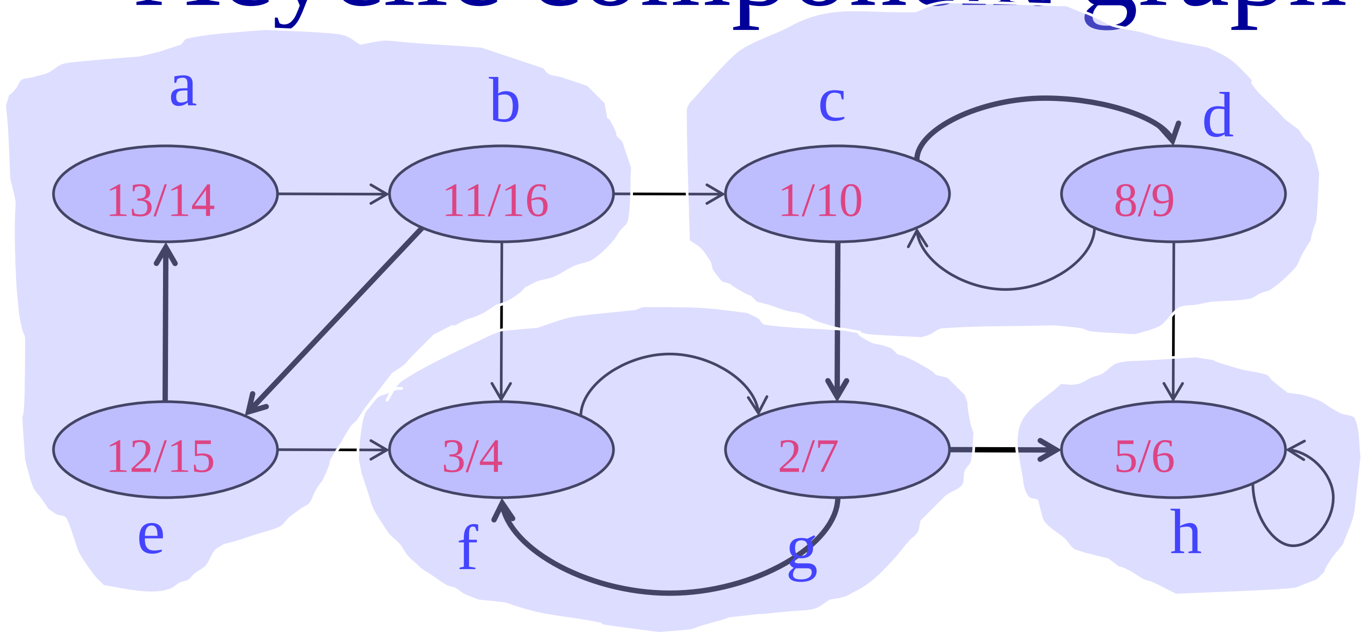


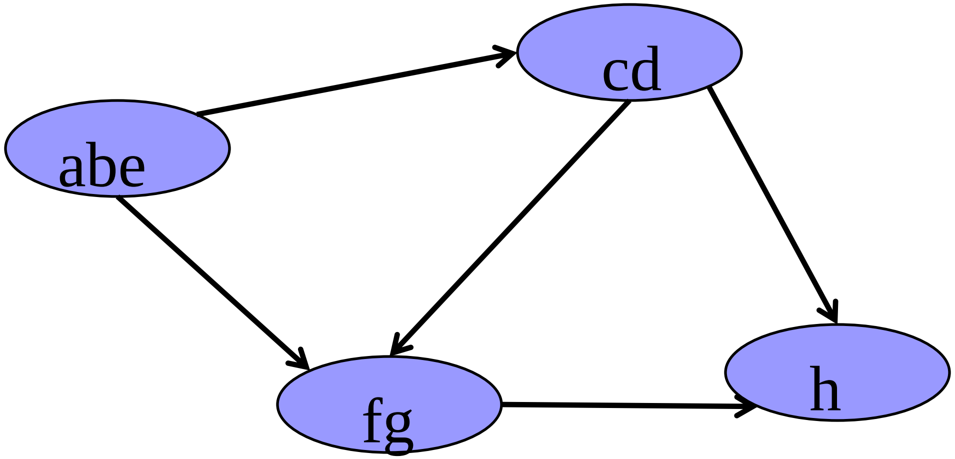


Digraph



Acyclic component graph





Theorem: In any DFS, all vertices in the same SCC are places in the same depth-first Tree.

Lemma: If two vertices in the same SCC, then no path between them ever leaves the SCC.

Def: **forefather**, $\varphi(u)$= vertex w : $u \rightarrow w$ and $f(w)$ is maximized.

- $f(u) \leq f(\varphi(u))$?
- $\varphi(\varphi(u)) = \varphi(u)$?

Theorem: In a digraph G, the forefather $\varphi(u)$ of any vertex u $\in$V in any DFS of G is an ancestor of u.

Col: In any DFS of a digraph G, verteicex u and $\varphi(u)$, $\forall$u $\in$V, lie in the same SCC.

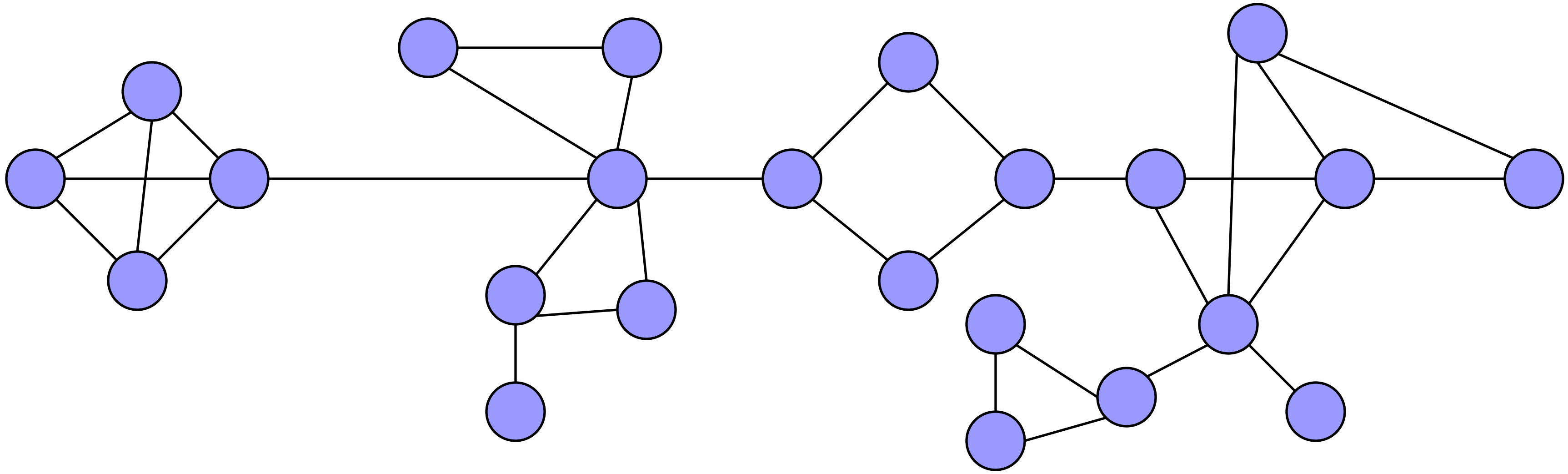
Theorem: In a digraph G, two vertices u,v $\in$V lie in the same SCC iff they have the same forefather in a DFS of G.

Theorem: The strongly connected components of a directed graph can be computed in time O(V+E).

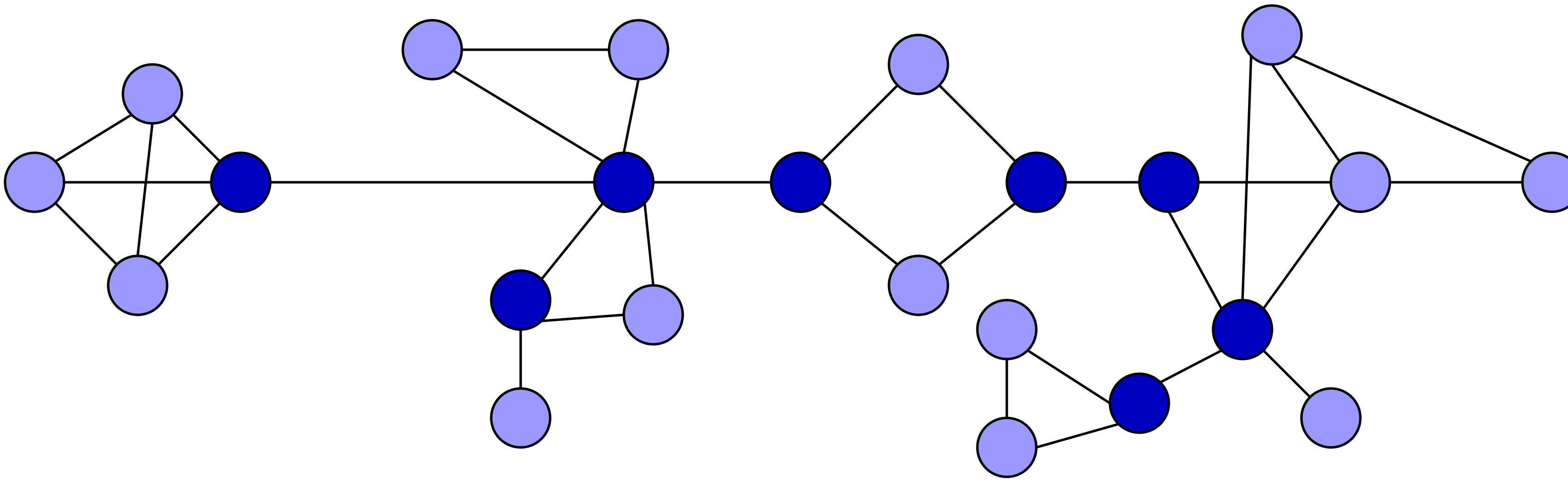
Biconnected Components

- An articulation point of a connected graph G is a vertex v such that $G - v$ is disconnected.
  - the deletion of v, together with all edges incident on v, produces a graph that has at least two connected components.
- A bridge of G is an edge whose removal disconnects G.
- A biconnected graph is a connected graph that has no articulation points.
  - every pair of vertices are connected by two vertex-disjoint paths.

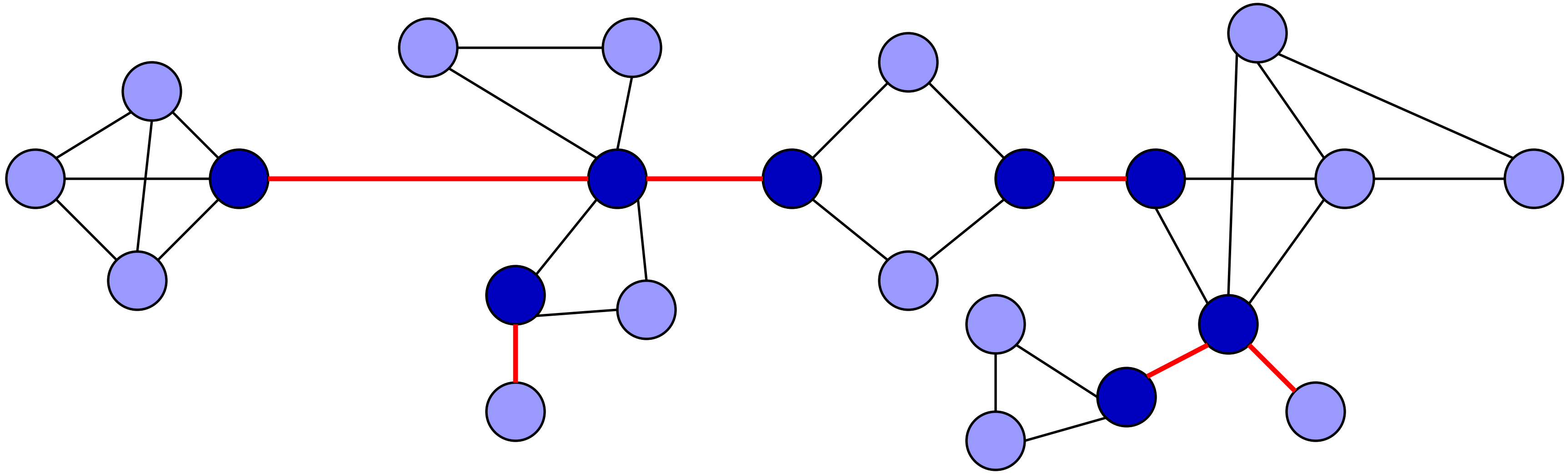
Example



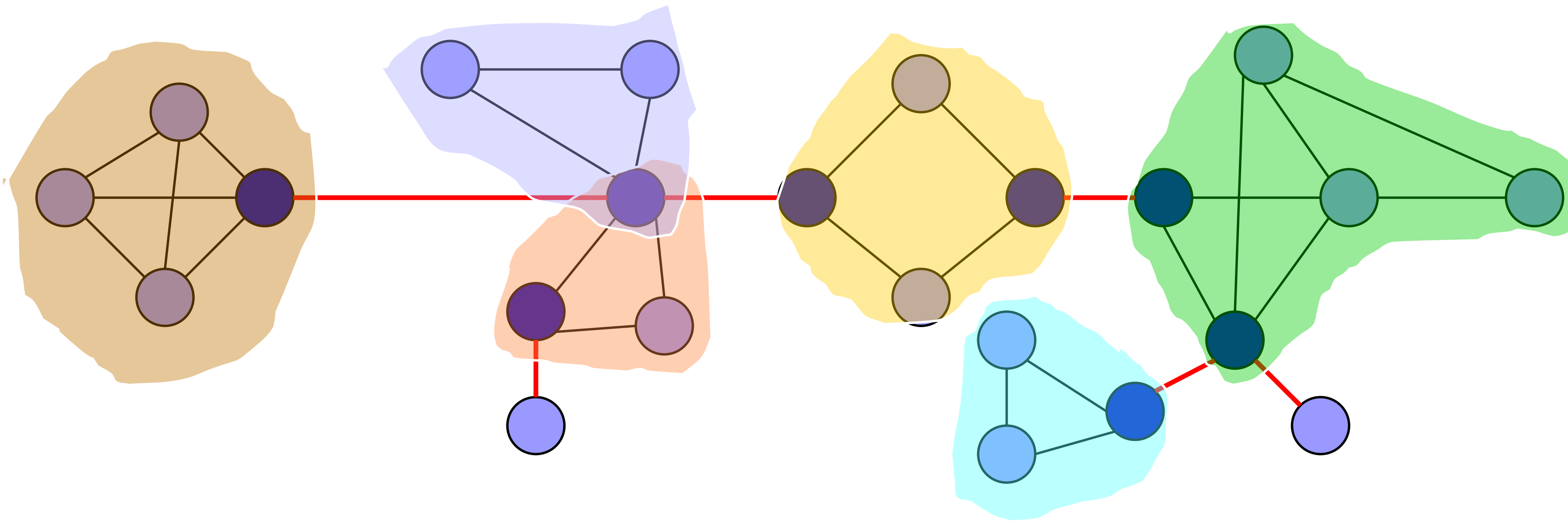
Example



Example



Example



Biconnected Components

- In many graph applications such as a communication network, articulation points are undesirable.
- Any graph other than K_2 is biconnected iff for any two distinct nodes u and v there are two vertex-disjoint paths from u to v.

Exercise:

- Finding biconnected components.
  - *Hint*: apply DFS.