Strongly Connected Components (SCC)

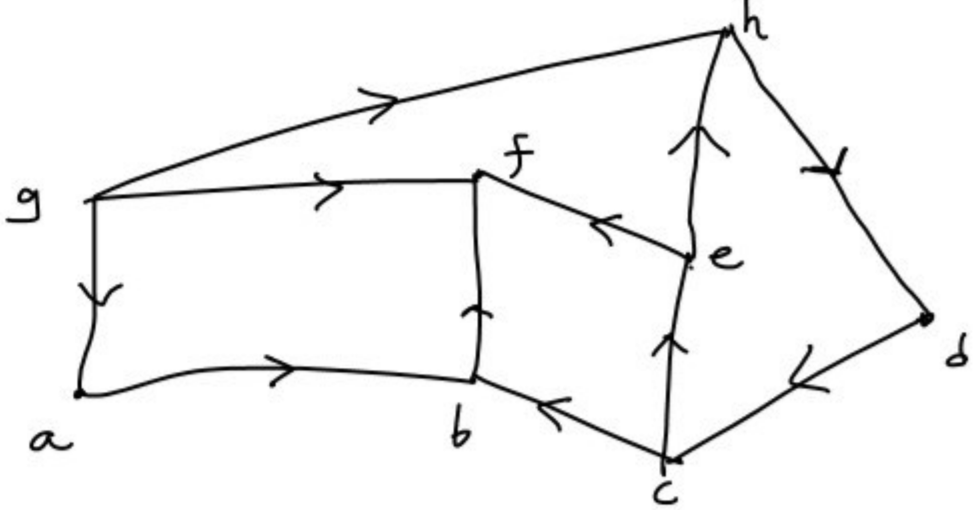
Defn: Let $G = (V, E)$ be a directed graph. Define a binary relation $\sim$ on V as follows: $x \sim y$ iff there is a (directed) path from x to y and there is a (directed) path from y to x

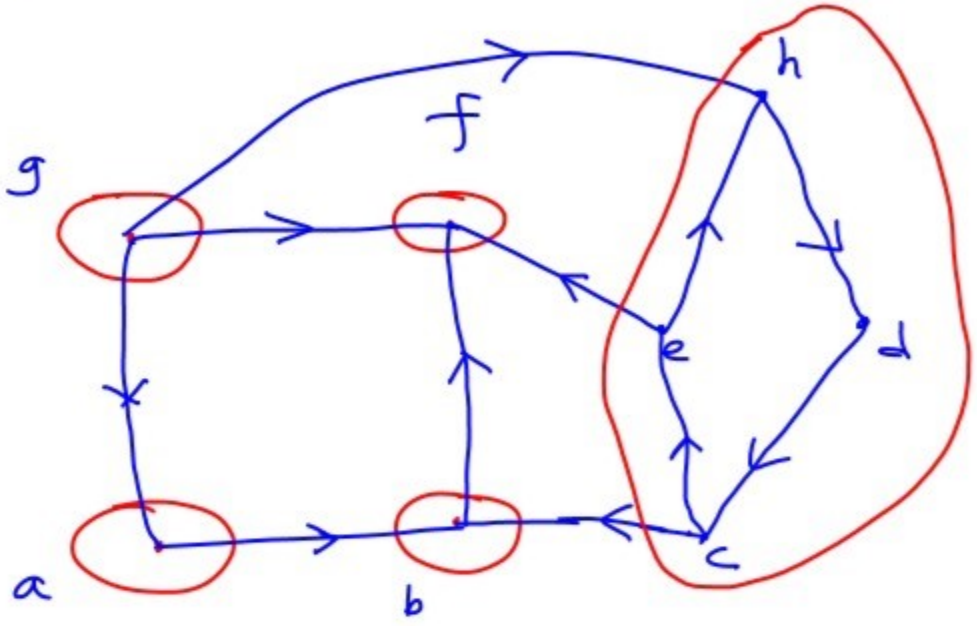
Exercise: Show that $\sim$ is an equivalence relation on V.

$\sim$ partitions V into equivalence class. These equivalence classes are strongly connected components of G.

$v \in V.$

$$\sim_v = \{ w \in V \mid v \sim w \}$$
$$= \{ w \in V \mid \text{there is a directed path from } v \text{ to } w \text{ and there is a directed path from } w \text{ to } v \}$$





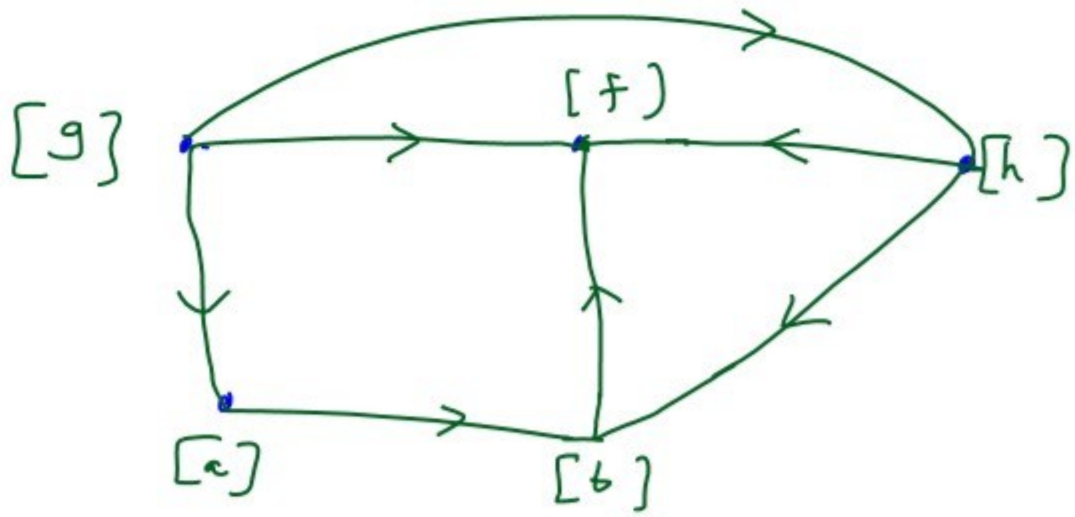
$\{a\}, \{b\}, \{c, e, d, h\}, \{f\}, \{g\}$

scc's

Definition $G^{scc} = (V^{scc}, E^{scc})$, $V^{scc} = \{\text{strongly connected components of } G\}$

$$E^{scc} = \{(C_1, C_2) \mid \exists x \in C_1 \ \exists y \in C_2 \ [(x,y) \in E]\}$$

(Strongly connected component graph of G)



$[v]$ is equivalence containing vertex v

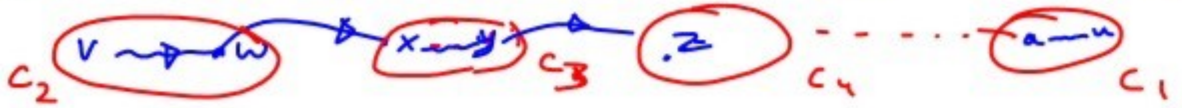
(for example $[h] = [c]$ etc.)

Lemma Let C_1, C_2 be two distinct components of $G=(V,E)$. If there are vertices $u \in C_1$, $v \in C_2$ s.t. $(u,v) \in E$ then there is no (directed) path from v to u in G.

Proof: If there was a path from v to u in G, then $u \sim v$. This contradicts that $u \in C_1$, $v \in C_2$. $\square$

Corollary: G^{scc} is acyclic.

Proof: Towards contradiction, assume that there is a cycle $C_1 C_2 \dots C_n C_1$ in G^{scc}. Then $\exists u \in C_1$, $\exists v \in C_2$ s.t. $(u,v) \in G$. If $w \in C_2$ and $x \in C_3$ s.t. $(w,x) \in E$ then there is a path from v to x.

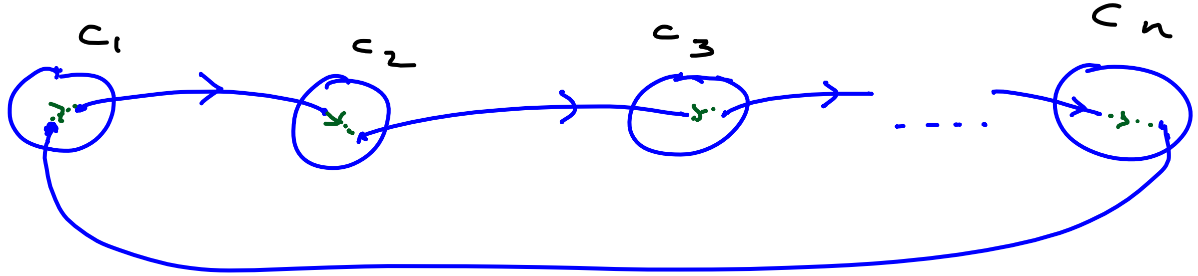


Continuing the argument

we get a path in G from v to u. This contradicts the previous Lemma. □

SCC's provide a decomposition of graph G.

Algorithms on G often, first divide G into SCC's and solve subproblems on SCC's and then combine the results to get solution for G.



⇒ all vertices on this cycle are in the same SCC

Contradiction

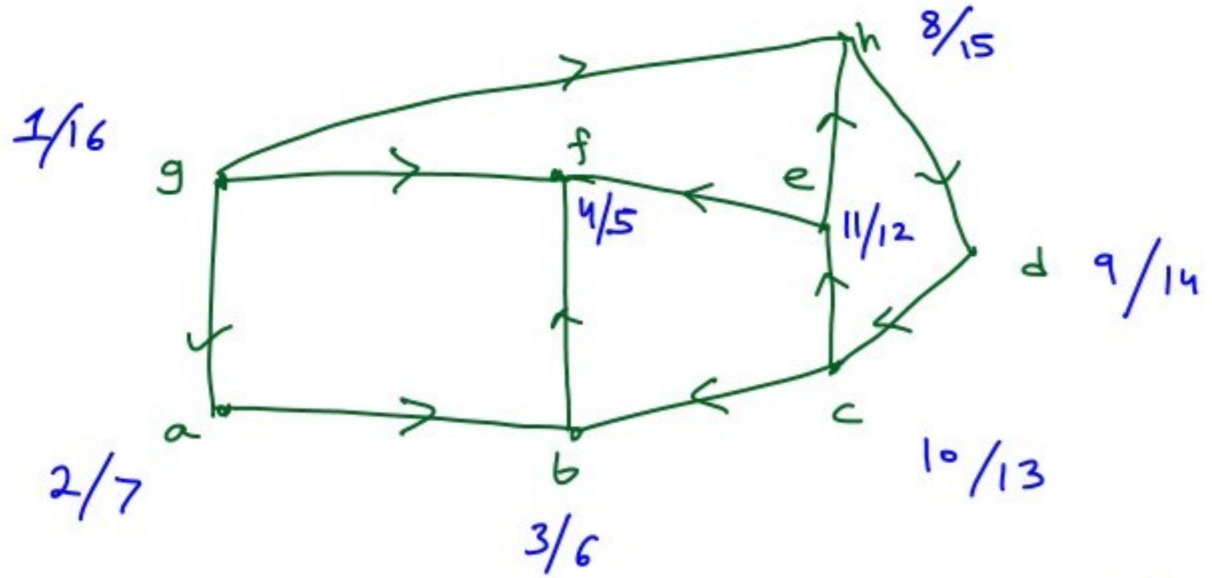
Algorithm to find SCC's of G

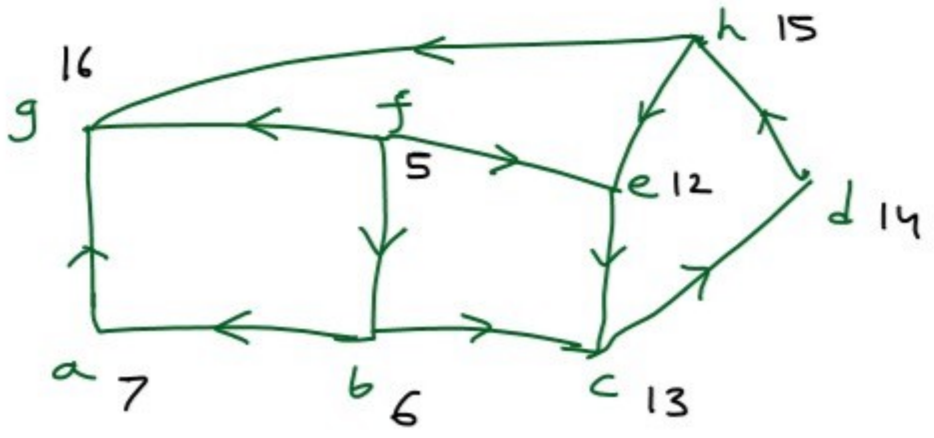
Find-SCC(G) // $G = (V, E)$

1. Call DFS(G) to find $u.f$ for all $u \in V$.
2. Compute G^T // G^T is obtained by reversing direction of edges in G.
3. Call DFS(G^T) [order of vertices considered in the main loop of DFS(G^T) is decreasing order of their finish time computed in line 1.]
4. Output vertices in each DFS tree (found by DFS(G^T)) as a separate strongly connected component.

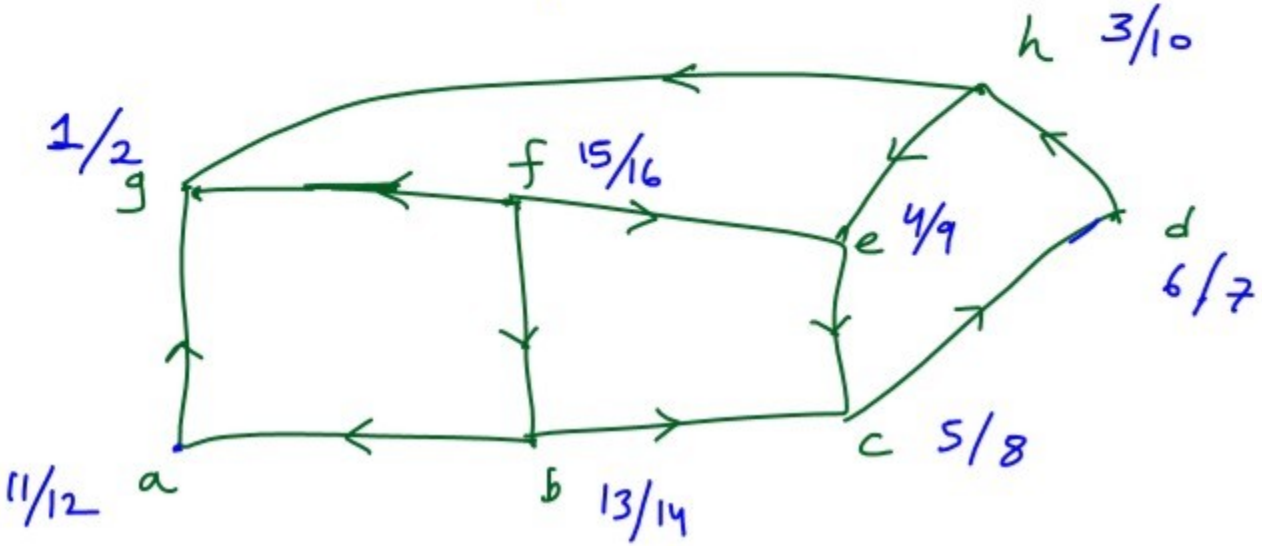
Complexity

$O(|V| + |E|)$





G^T



DFS (G^T)

{g}, {h, e, c, d}, {a}

{b}, {f}

Correctness of SCC algorithm

Defn Let C be a connected component of G.
(SCC)

$$f(C) = \max_{v \in C} \{ v.f \}$$

Lemma: Let C_1, C_2 be two distinct strongly connected Comp. of G.
If $(u,v) \in E(G)$ s.t. $u \in C_1$ and $v \in C_2$ then

$$f(C_1) > f(C_2).$$

Proof: Consider DFS(G) in step 1 of the algorithm.
Let 'a' be the first vertex in $C_1 \cup C_2$ visited in this DFS.

Case I $\quad a \in C_1$

When vertex 'a' is discovered then there is a white path from 'a' to any vertex in $C_1 \cup C_2$. By white path theorem each vertex in $C_1 \cup C_2$ is a descendent of 'a'.

Let b be the first vertex in C_2 which is discovered while a is grey. By white path theorem again every vertex in C_2 is a descendent of b. For all $w \in C_2$, $w \neq b$, $w.f < b.f$, $f(C_2) = b.f$.

As b is a descendent of a, $a.f > b.f$.

$$f(C_1) \geq a.f > b.f = f(C_2)$$

Case II $\quad a \in C_2.$

By white path theorem, every vertex in C_2 is a descendent of a. $\quad f(C_2) = a.f$.

No vertex in C_1 is a descendent of 'a'.

$\Rightarrow$ Any vertex in C_1 is explored after $a.f$.

$$\Rightarrow \quad f(C_1) > f(C_2)$$

$\square$

Observation strongly connected components in G and G^T are the same.

Exercise: prove this observation.

Corollary: Let (u,v) be an edge in G^T, s.t. $u \in C_1$, $v \in C_2$, $C_1 \neq C_2$. (C_1, C_2 are strongly connected components of G or G^T). Then $f(C_1) < f(C_2)$.

Proof: $(u,v) \in G^T \implies (v,u) \in G$.

$\implies f(C_2) > f(C_1)$ (By previous Lemma) $\square$

Let us consider first DFS tree produced by scc algorithm.

Let v be the vertex with largest $v.f$ in G,

All vertices in $[v]$ are descendants of v by white path theorem.

If a vertex $w \in C$ $(C \neq [v])$ is discovered during DFS-visit(G^T, u), then w.l.o.g. let w be the first such vertex.

$\implies$ there is a vertex x in $[v]$ s.t. $(x,w) \in E(G^T)$

$\implies f([w]) > f([x]) = f([v])$

Contradiction by our choice of v.

Lemma: Algorithm Find-SCC(G) correctly outputs the strongly connected components of G.

Proof: By induction on k, we show that vertices of k^{th} DFS tree found in DFS(G^T) is a strongly connected component of G^T.

Base Case: $k = 1$. We argued this just before this Lemma.

Induction Step: Suppose first k_0 SCC's have been output. Consider $(k_0+1)^{th}$ DFS tree. Assume its search starts with vertex u.

$u.f > w.f$ where w $(w \neq u)$ is not visited by DFS(G^T) yet.

$$f([u]) = u.f$$

By white path theorem, every vertex in $[u]$ is a descendent of u.

If there is an edge $(y\ z)$ in G^T s.t. $y \in [u]$ and $z \notin [u]$

then $f([z]) > f([y]) = f([u])$

$\Rightarrow$ $[z]$ is not the scc not output ($\because$ then $f([z]) < f([u])$, by our choice of u)

By our choice of u, only vertices outside $[u]$ reachable in $DFS(G^T, u)$ are one's which have already been visited by DFS. So these vertices are not part of $(k_0+1)^{th}$ DFS tree.

$\Rightarrow$ $(k_0+1)^{th}$ DFS tree has exactly $[u]$ as its set of vertices.

(SCC are output in decreasing order of the $f(\cdot)$ values)

$\square$