Strongly Connected Components

Dr. G P Gupta

1

Strongly Connected Components

Definition: a strongly connected component (SCC) of a directed graph G=(V,E) is a maximal set of vertices $U \subseteq V$ such that

- For each $u,v \in U$ we have both $u \mapsto v$ and $v \mapsto u$

i.e., u and v are mutually reachable from each other ($u \leftrightarrows v$)

Let $G^T=(V,E^T)$ be the ***transpose*** of G=(V,E) where

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

- i.e., E^T consists of edges of G with their directions reversed

Constructing G^T from G takes O(V+E) time (adjacency list rep)

Note: G and G^T have the same SCCs ($u \leftrightarrows v$ in G $\Leftrightarrow$ $u \leftrightarrows v$ in G^T)

2

Strongly Connected Components

Algorithm

(1) Run **DFS**(G) to compute finishing times for all $u \in V$

(2) Compute G^T

(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[u] computed in Step (1)

(4) Output vertices of each DFT in DFF of Step (3) as a separate SCC

3

Strongly Connected Components

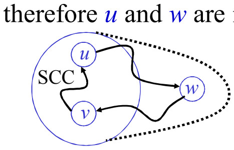
Lemma 1: no path between a pair of vertices in the same SCC, ever leaves the SCC

Proof: let u and v be in the same SCC $\Rightarrow u \leftrightarrows v$

let w be on some path $u \mapsto w \mapsto v \Rightarrow$ $\boxed{u \mapsto w}$

but $v \mapsto u \Rightarrow \exists$ a path $w \mapsto v \mapsto u \Rightarrow$ $\boxed{w \mapsto u}$

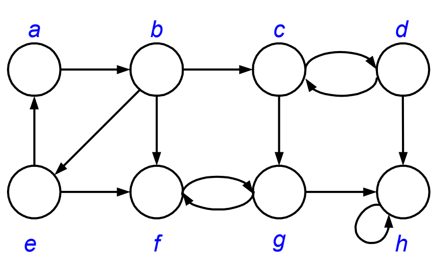
therefore u and w are in the same SCC



QED

4

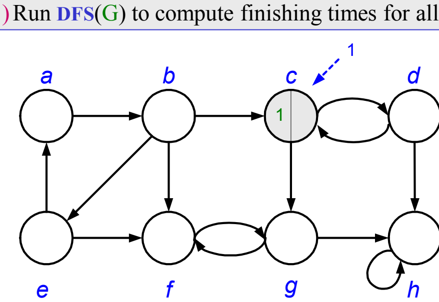
SCC: Example



5

SCC: Example

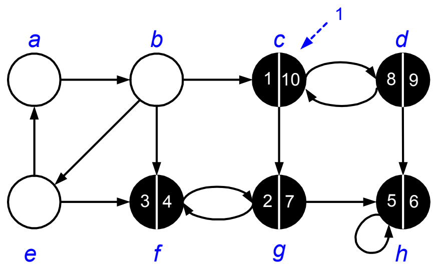
(1) Run **DFS**(G) to compute finishing times for all $u \in V$



6

SCC: Example

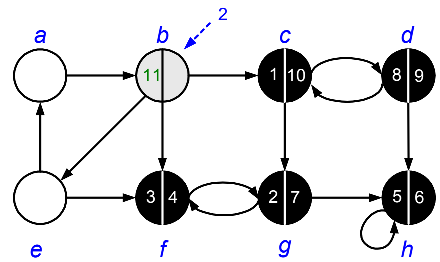
(1) Run **DFS**(G) to compute finishing times for all $u \in V$



7

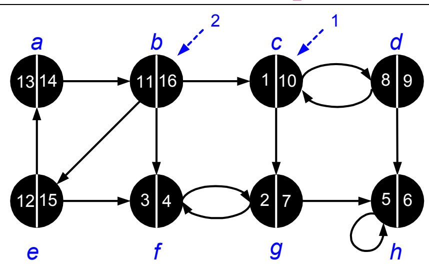
SCC: Example

(1) Run **DFS**(G) to compute finishing times for all $u \in V$



8

SCC: Example



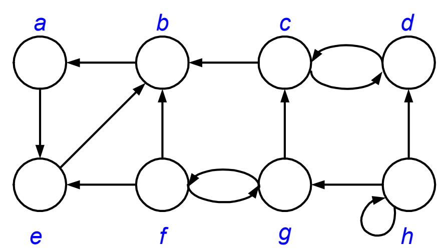
Vertices sorted according to the finishing times:

⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩

9

SCC: Example

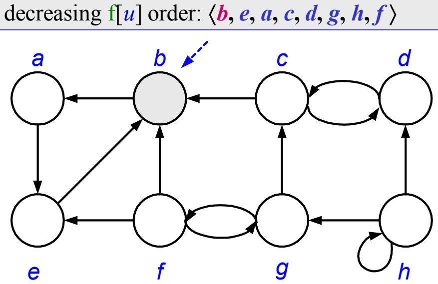
(2) Compute G^T



10

SCC: Example

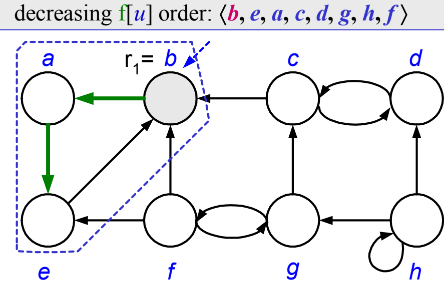
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



11

SCC: Example

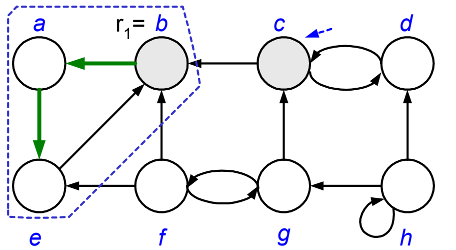
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



12

SCC: Example

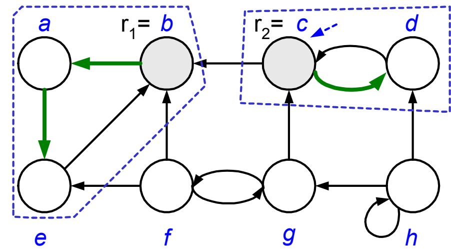
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



13

SCC: Example

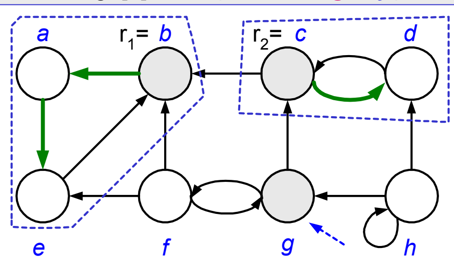
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



14

SCC: Example

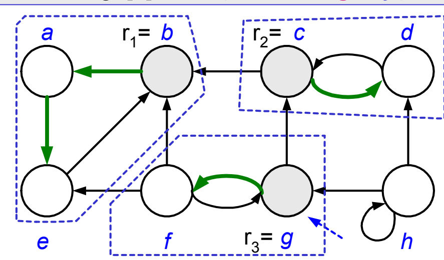
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



15

SCC: Example

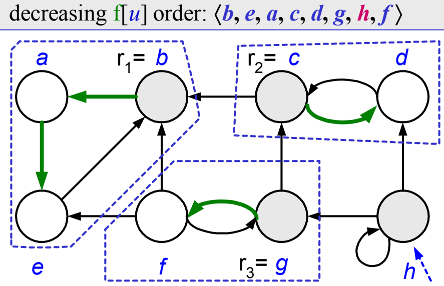
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



16

SCC: Example

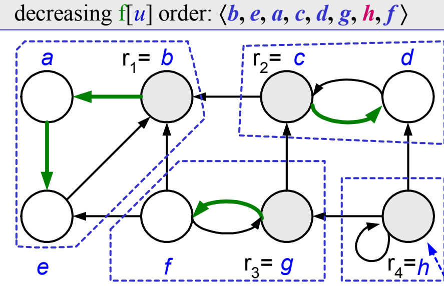
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



17

SCC: Example

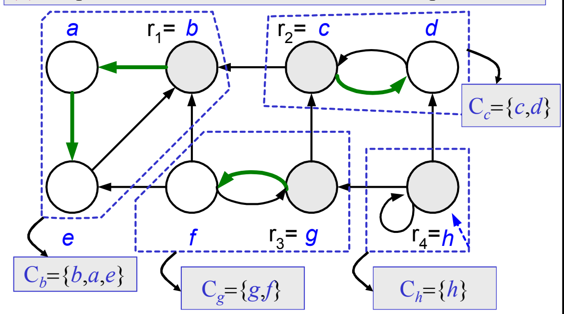
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



18

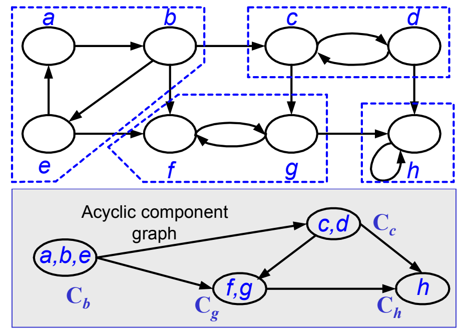
SCC: Example

(4) Output vertices of each DFT in DFF as a separate SCC



19

SCC: Example



20