APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

STUDY MATERIALS

KTU ASSIST

Strongly Connected Components

Strongly Connected Components

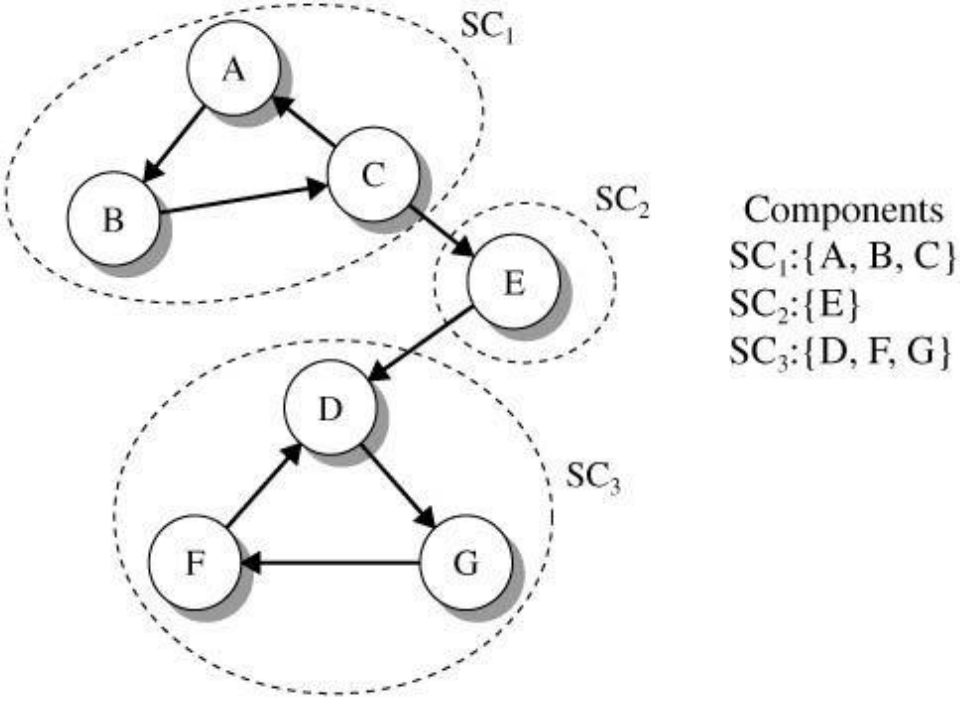
Definition: Strongly connected component (SCC) of a directed graph $G=(V,E)$ is a maximal set of vertices $C \subseteq V$ such that

- For every pair of vertices u and v in C (u ~> v and v~>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

Strongly Connected Components

- Any graph can be partitioned into a unique set of strong components.



Strongly connected components in a directed graph.

Strongly Connected Components

Algorithm

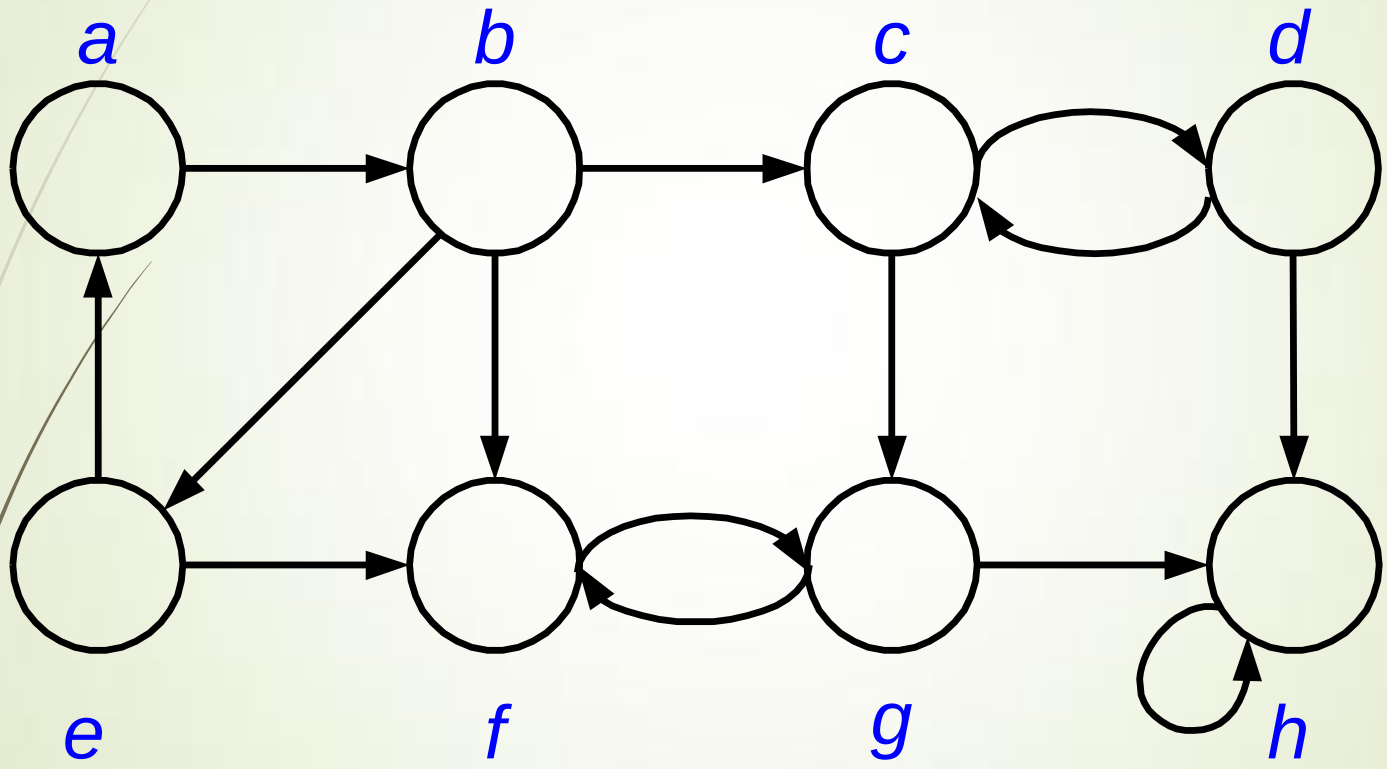
(1) Run **DFS**(G) to compute finishing times for all $u \in V$

(2) Compute G^T

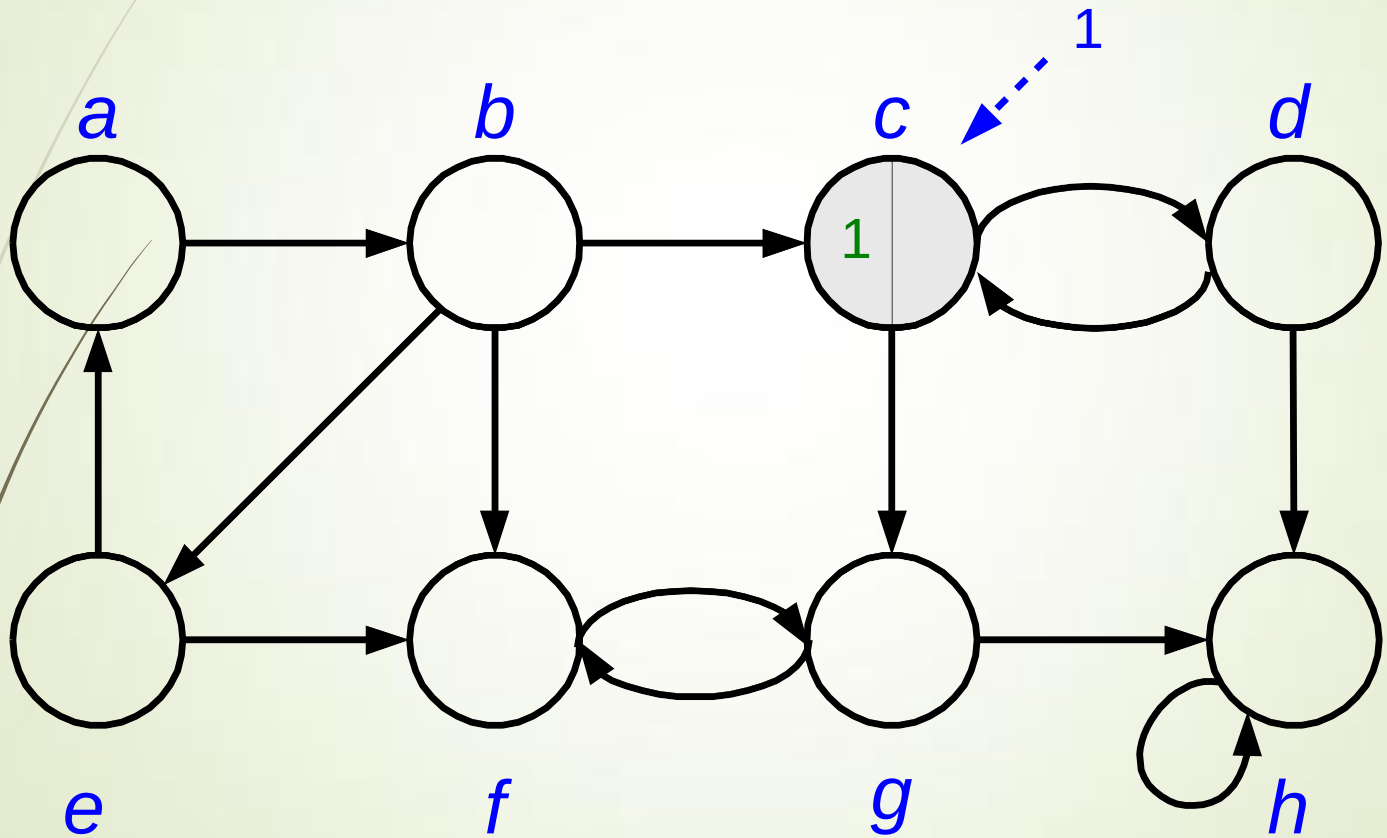
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[u] computed in Step (1)

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

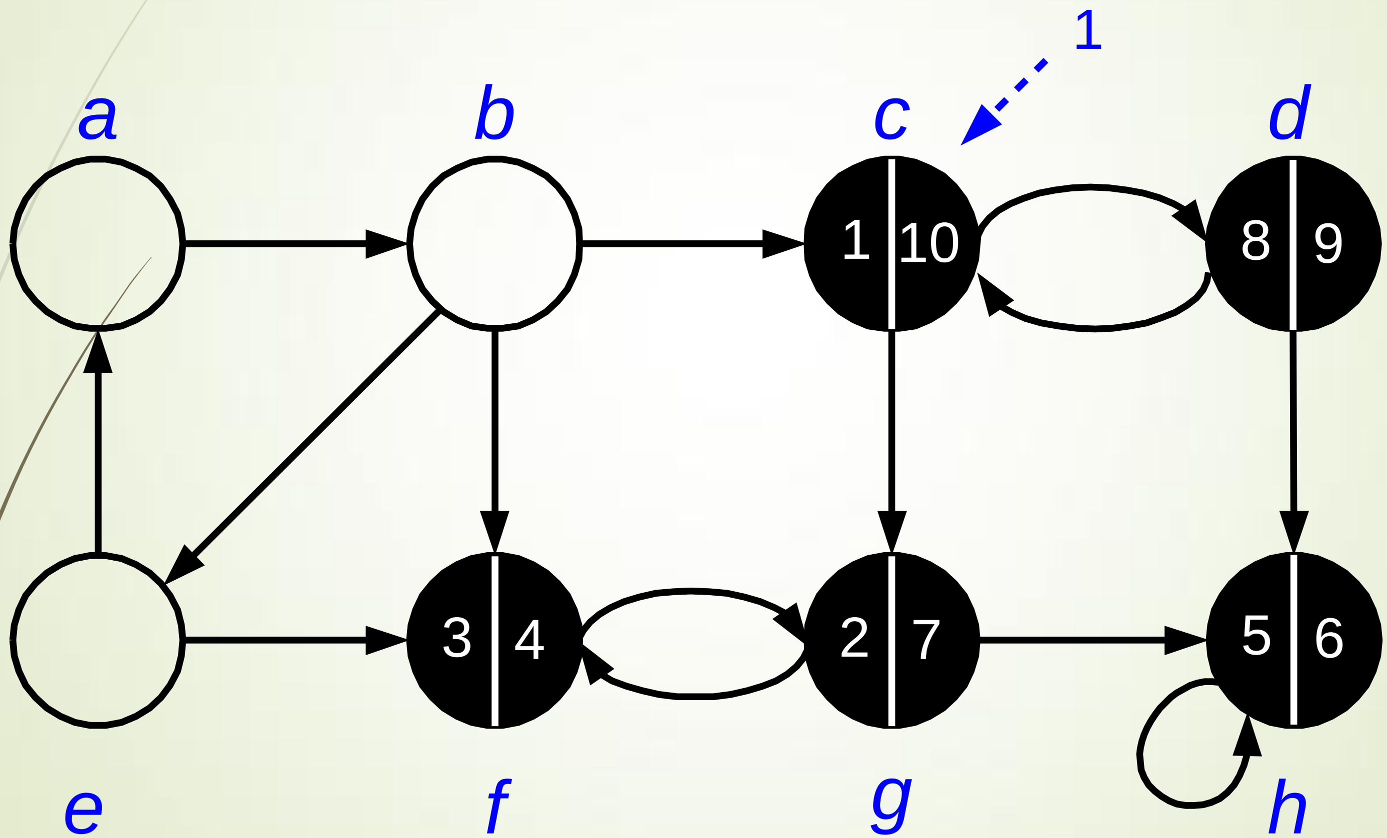
SCC: Example



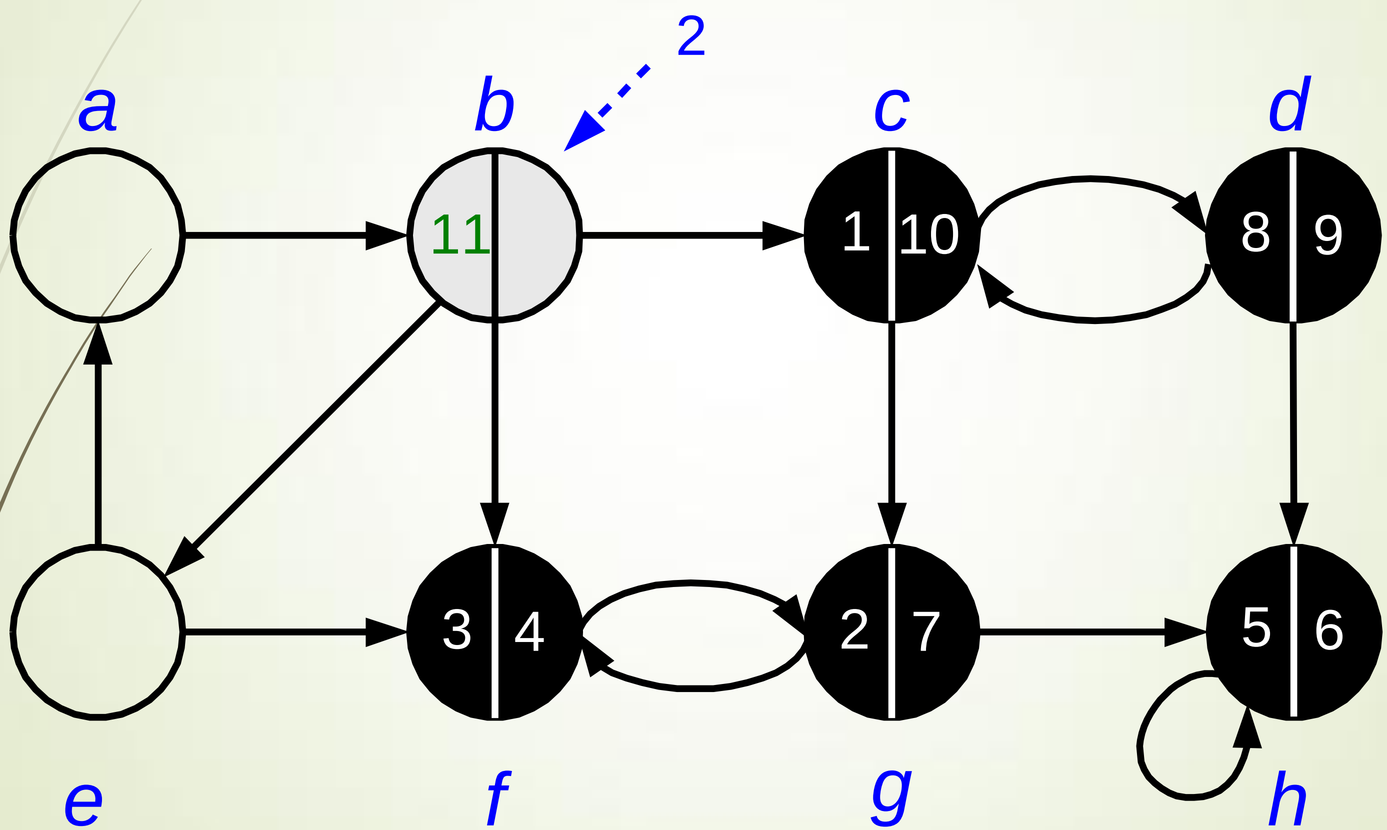
(1) Run **DFS**(G) to compute finishing times for all $u \in V$

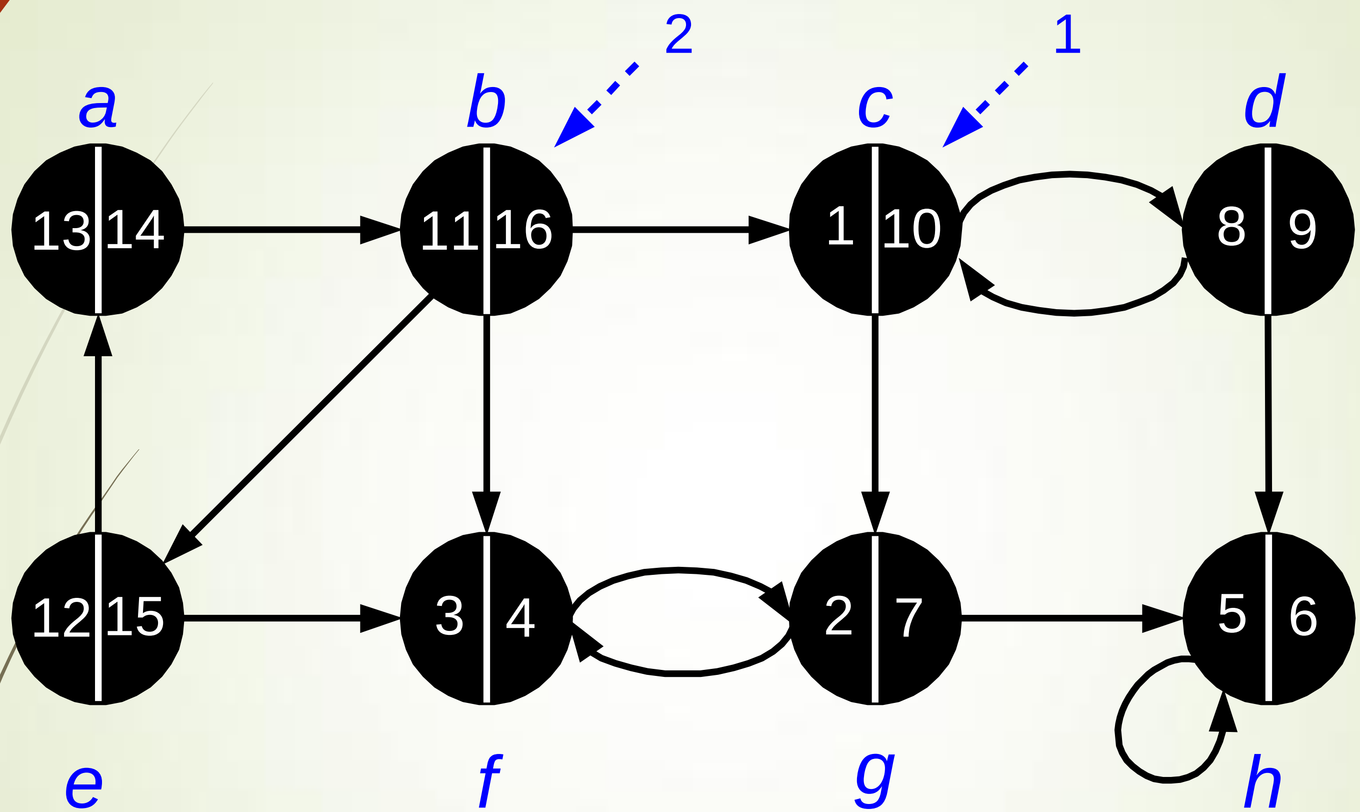


(1) Run **DFS**(G) to compute finishing times for all $u \in V$



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

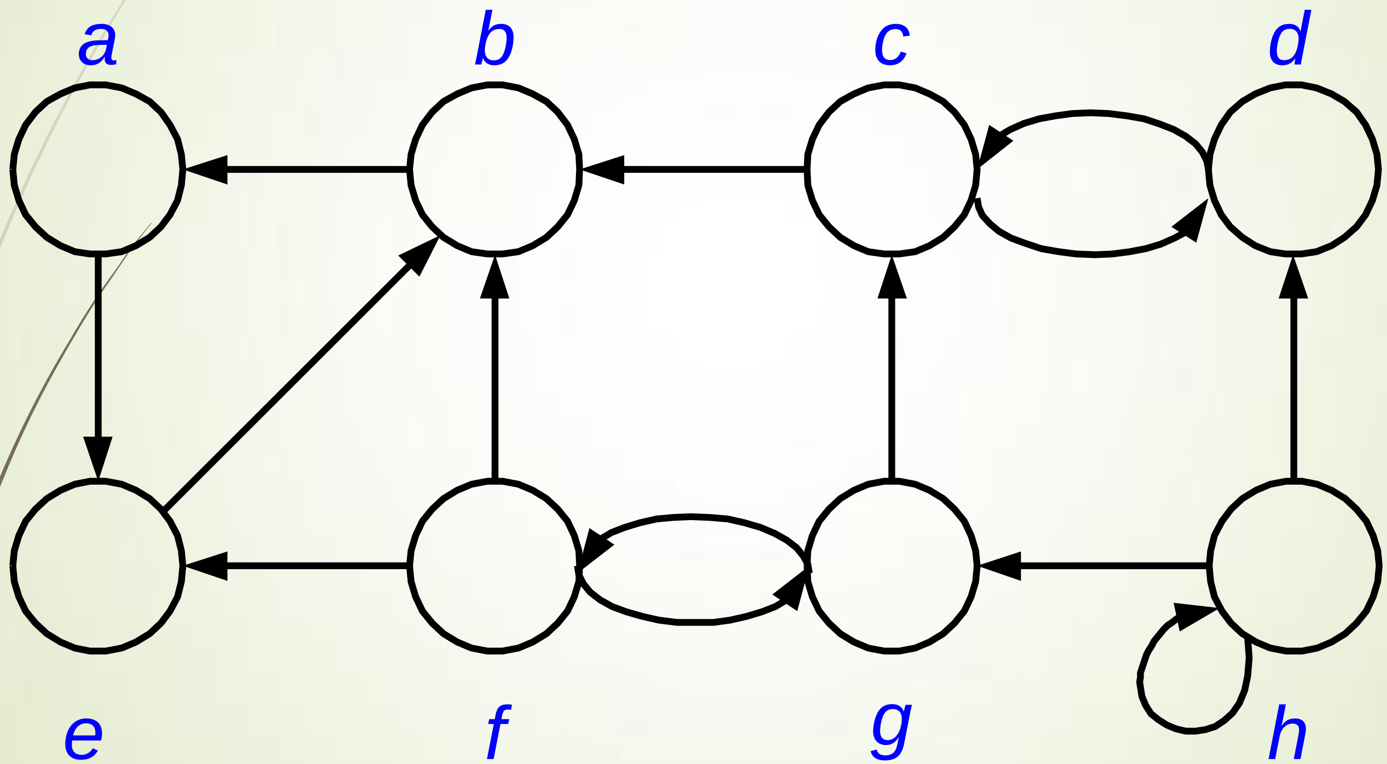




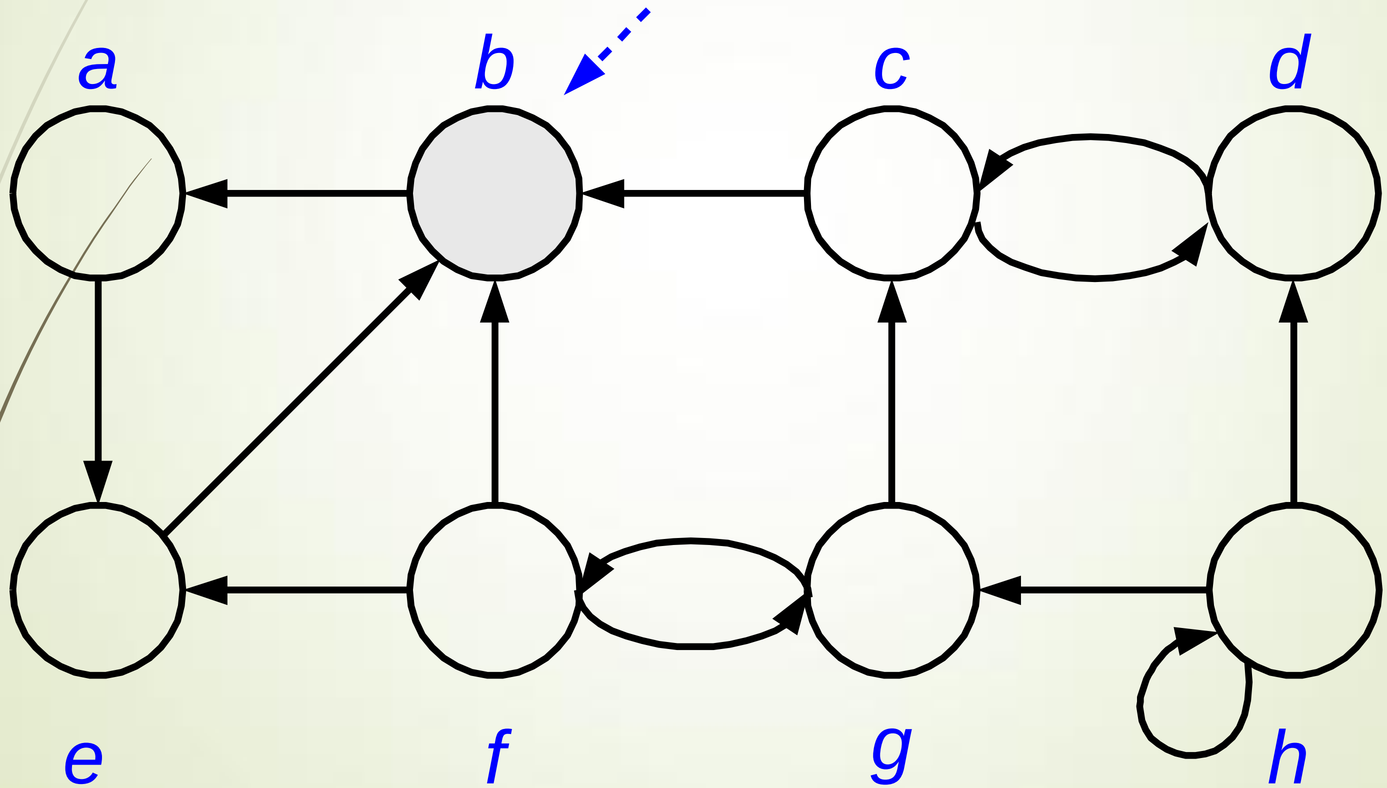
Vertices sorted according to the finishing times:

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

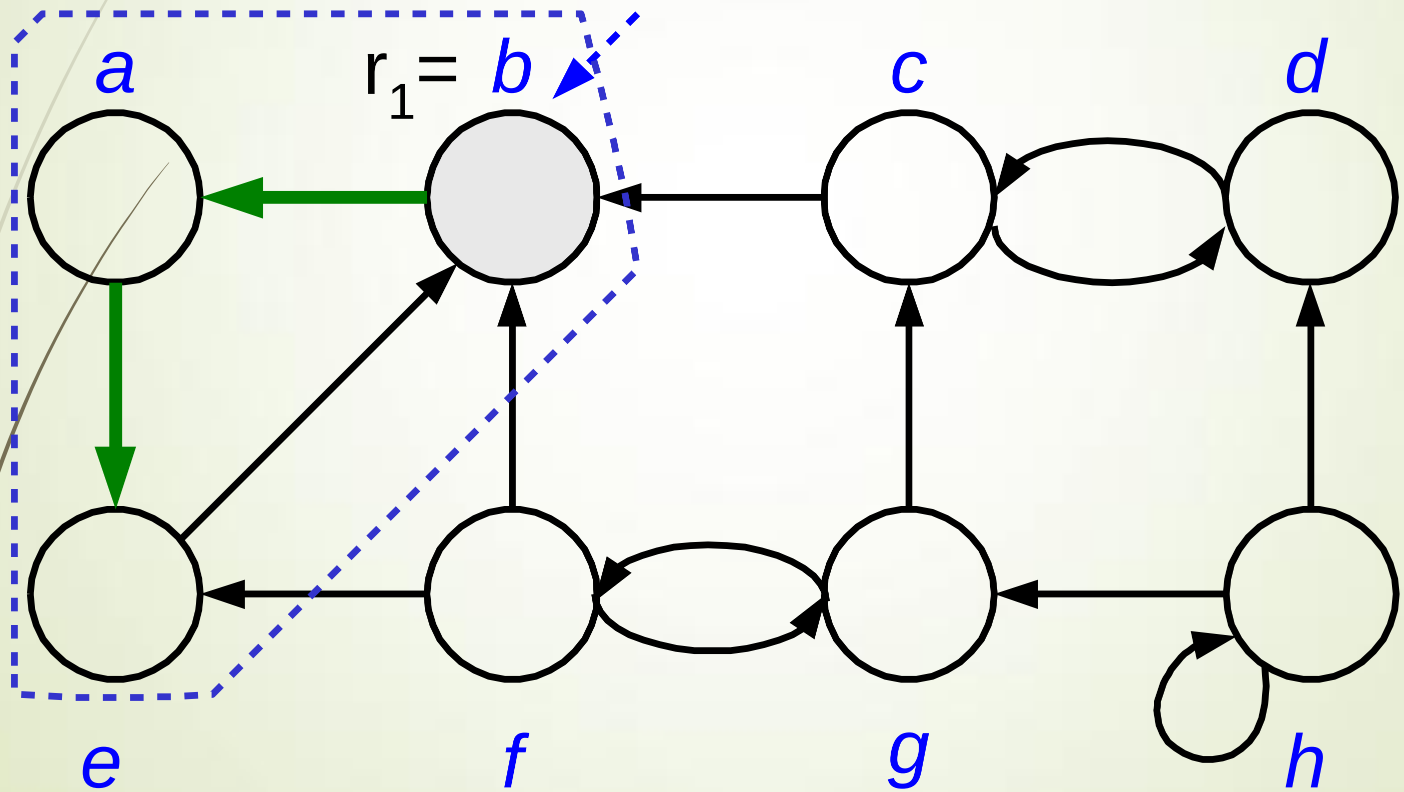
(2) Compute G^T



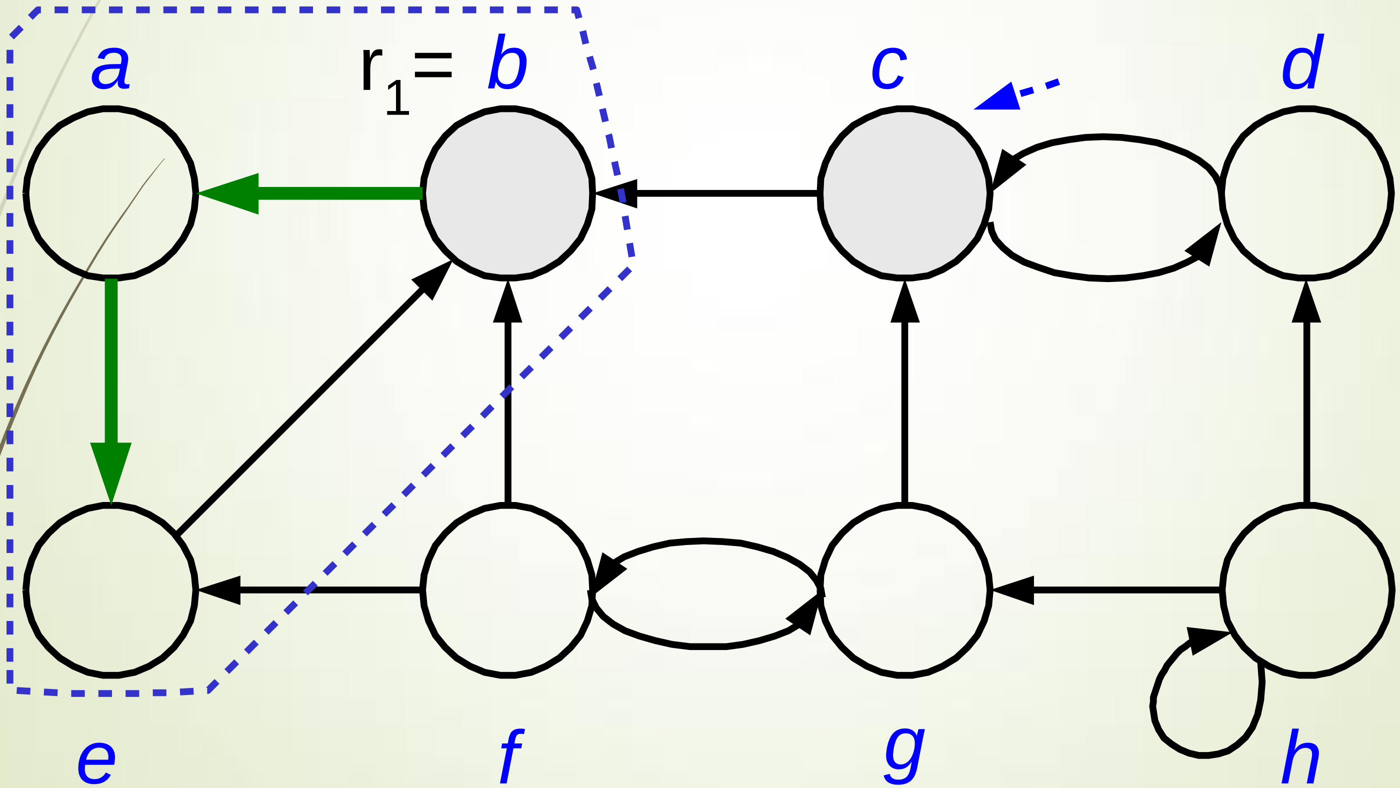
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing $f[u]$ order: $\langle$***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f*** $\rangle$



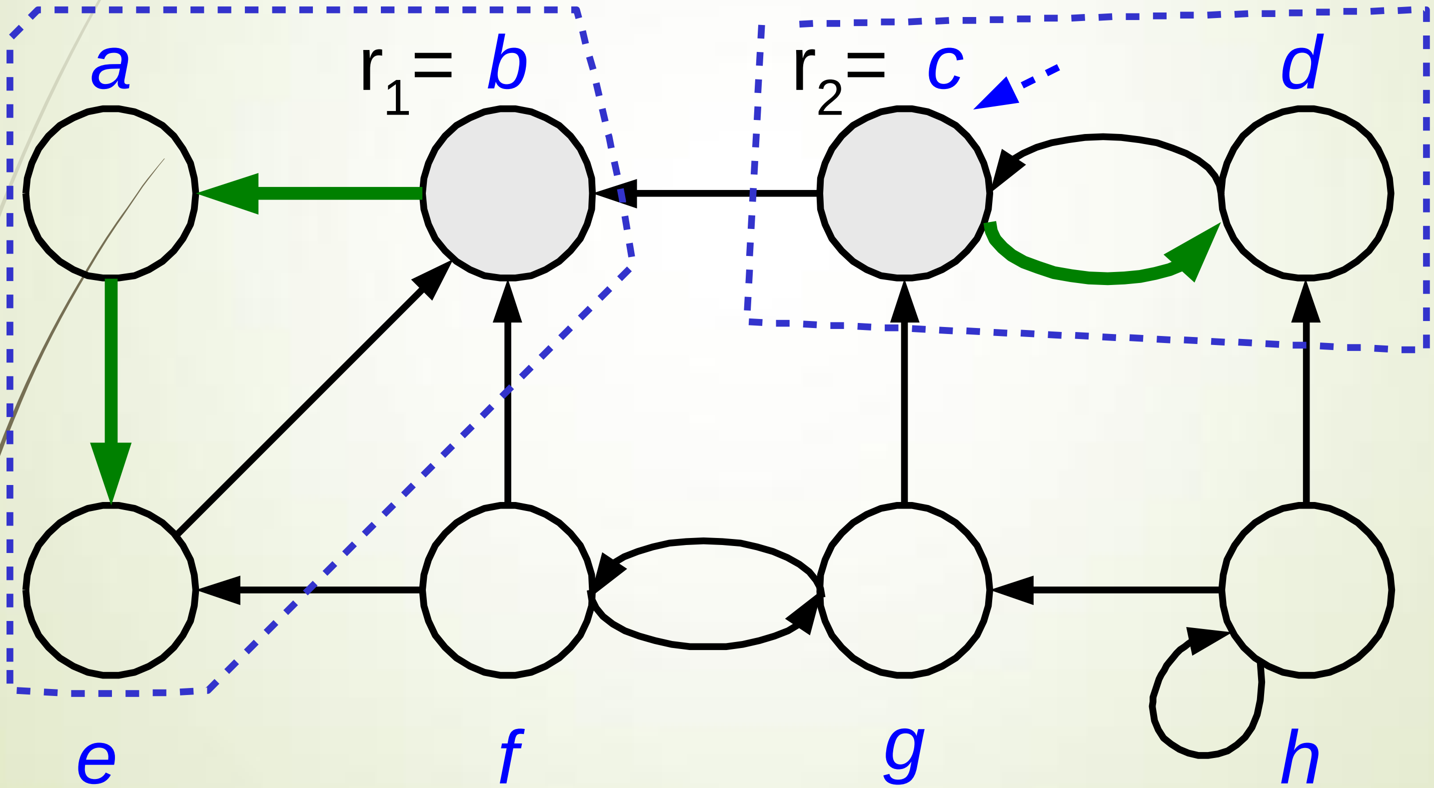
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[u] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f*** ⟩



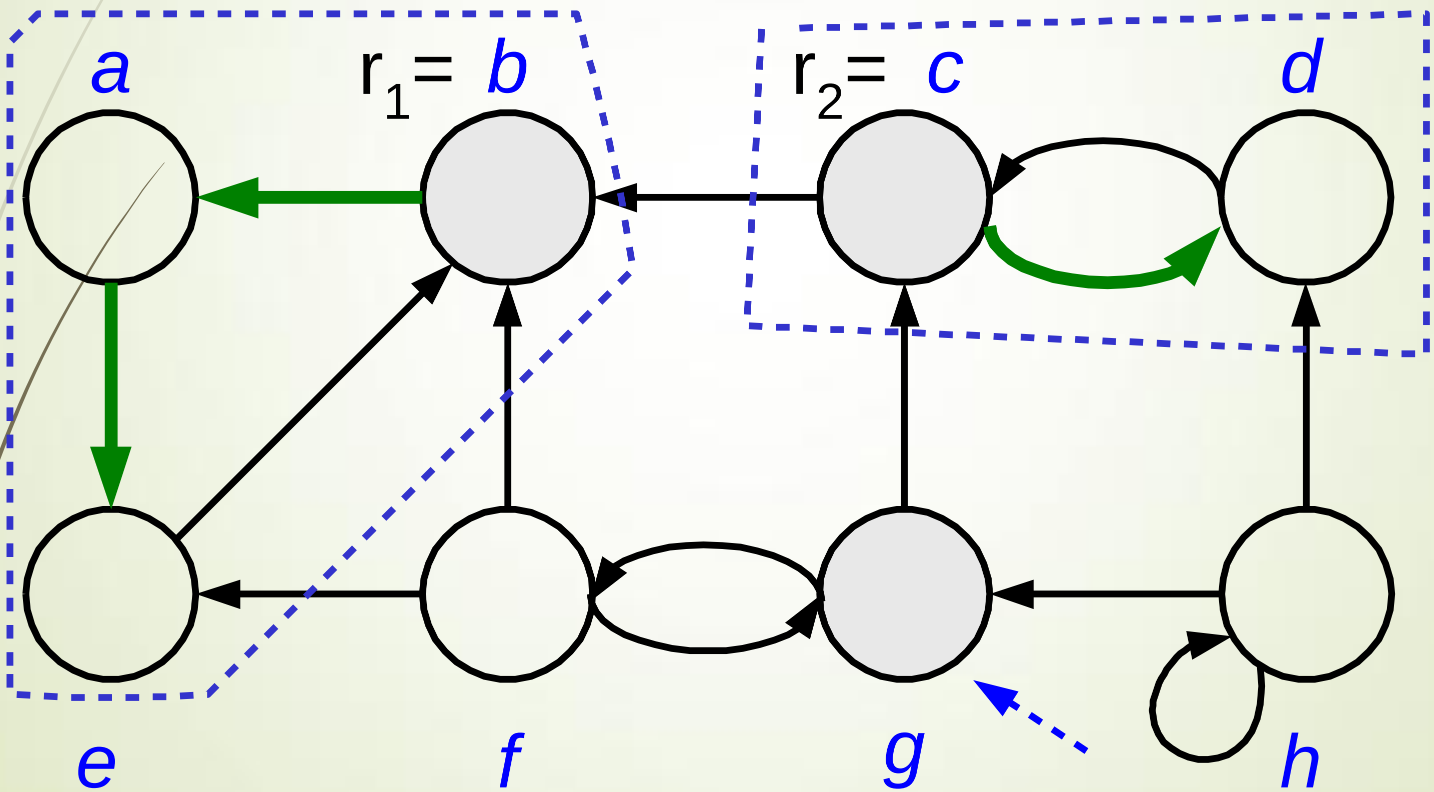
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f*** ⟩



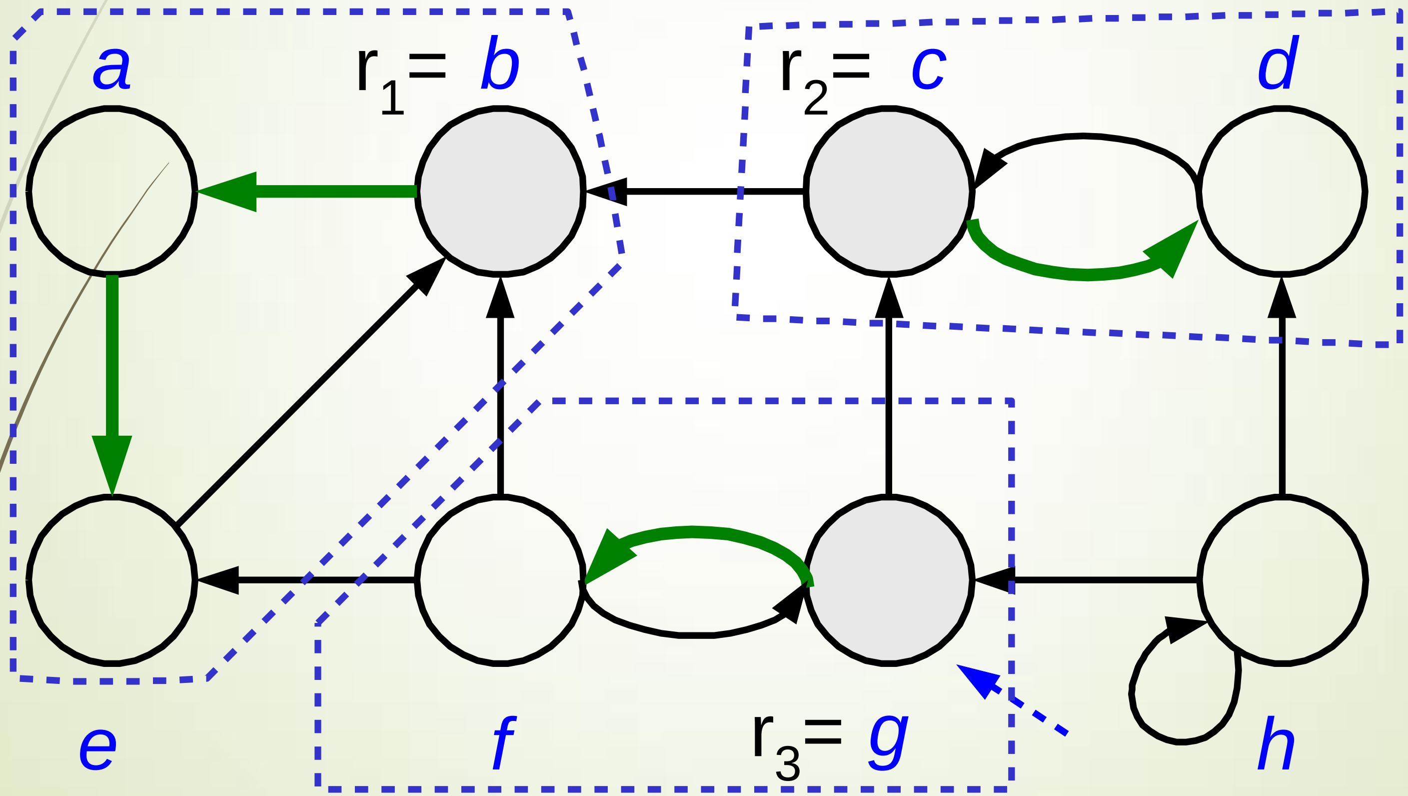
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f*** ⟩



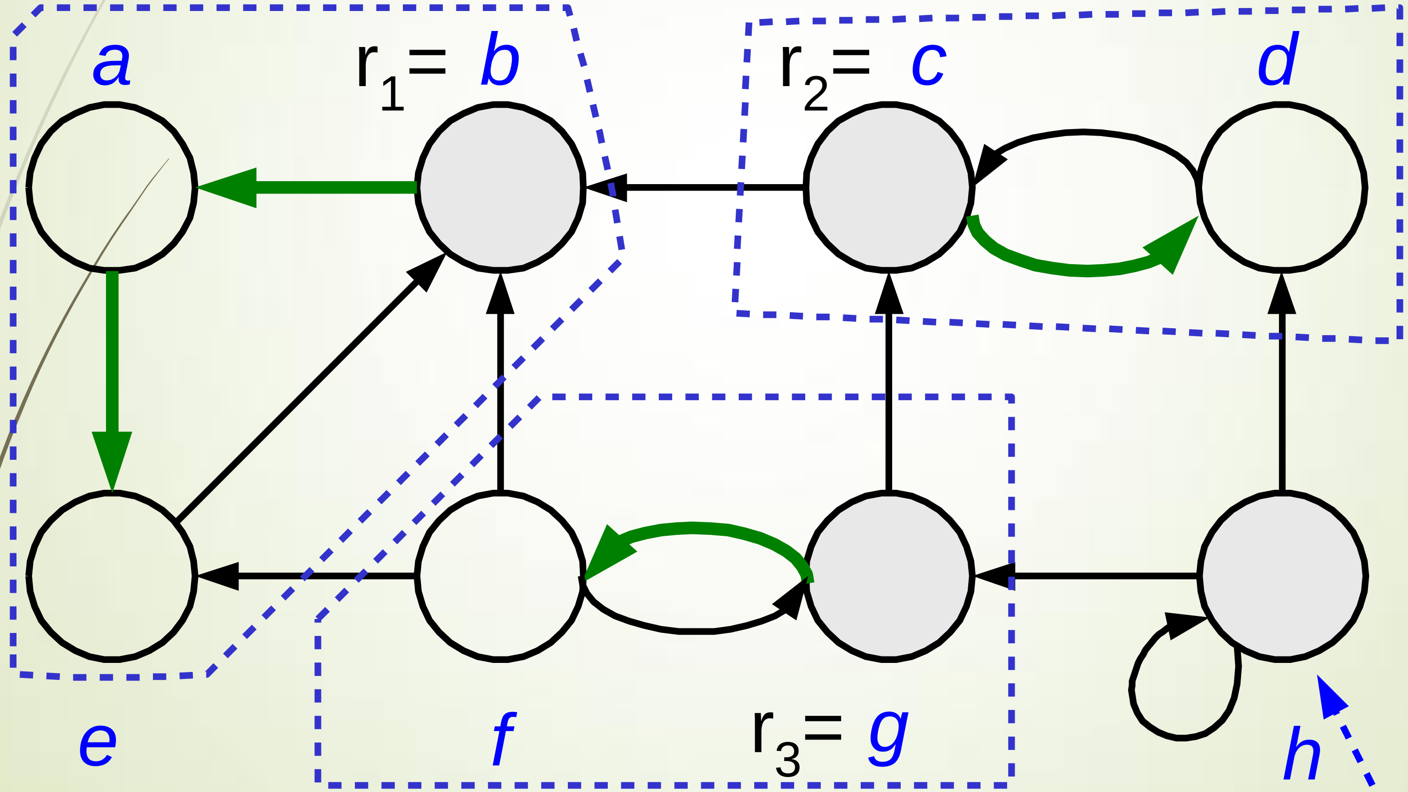
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩



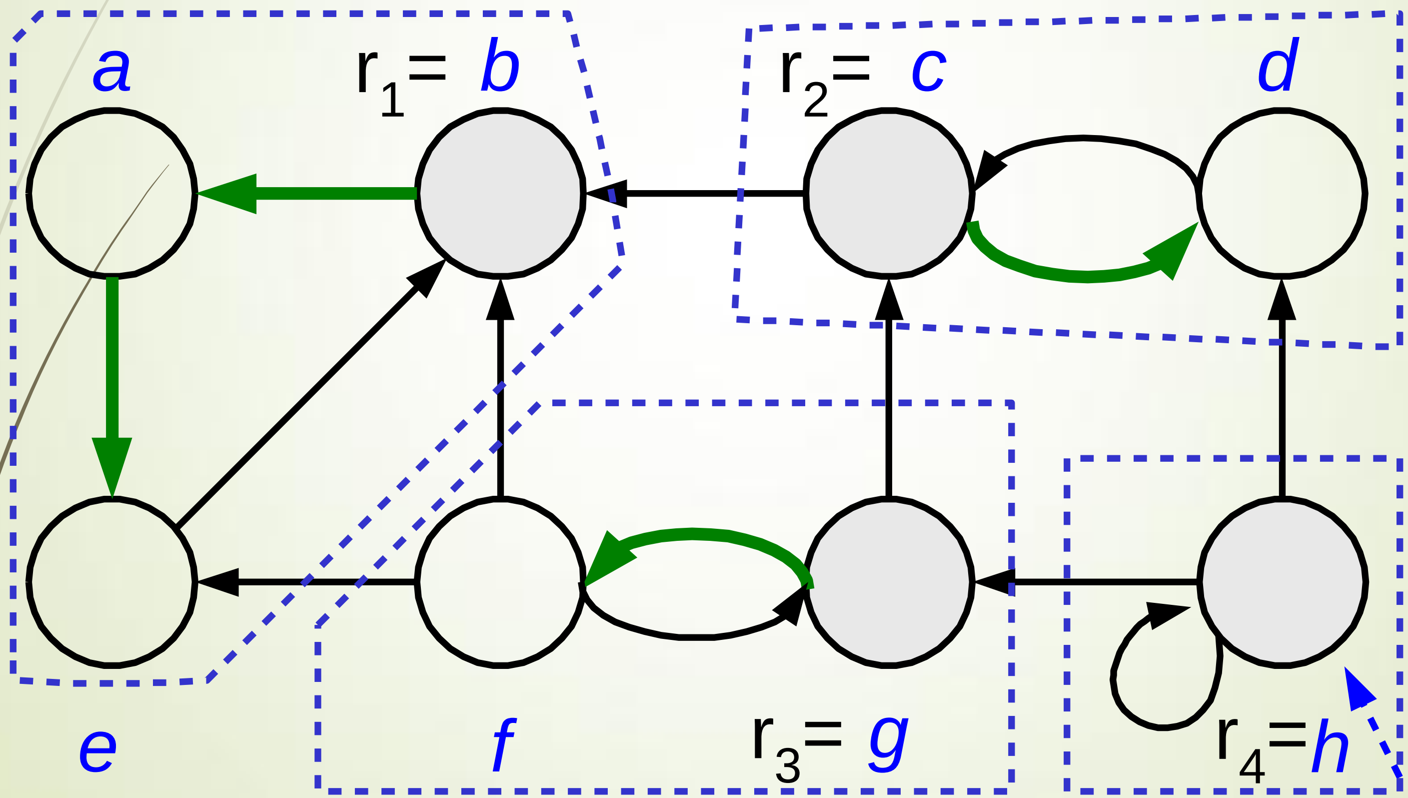
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f*** ⟩



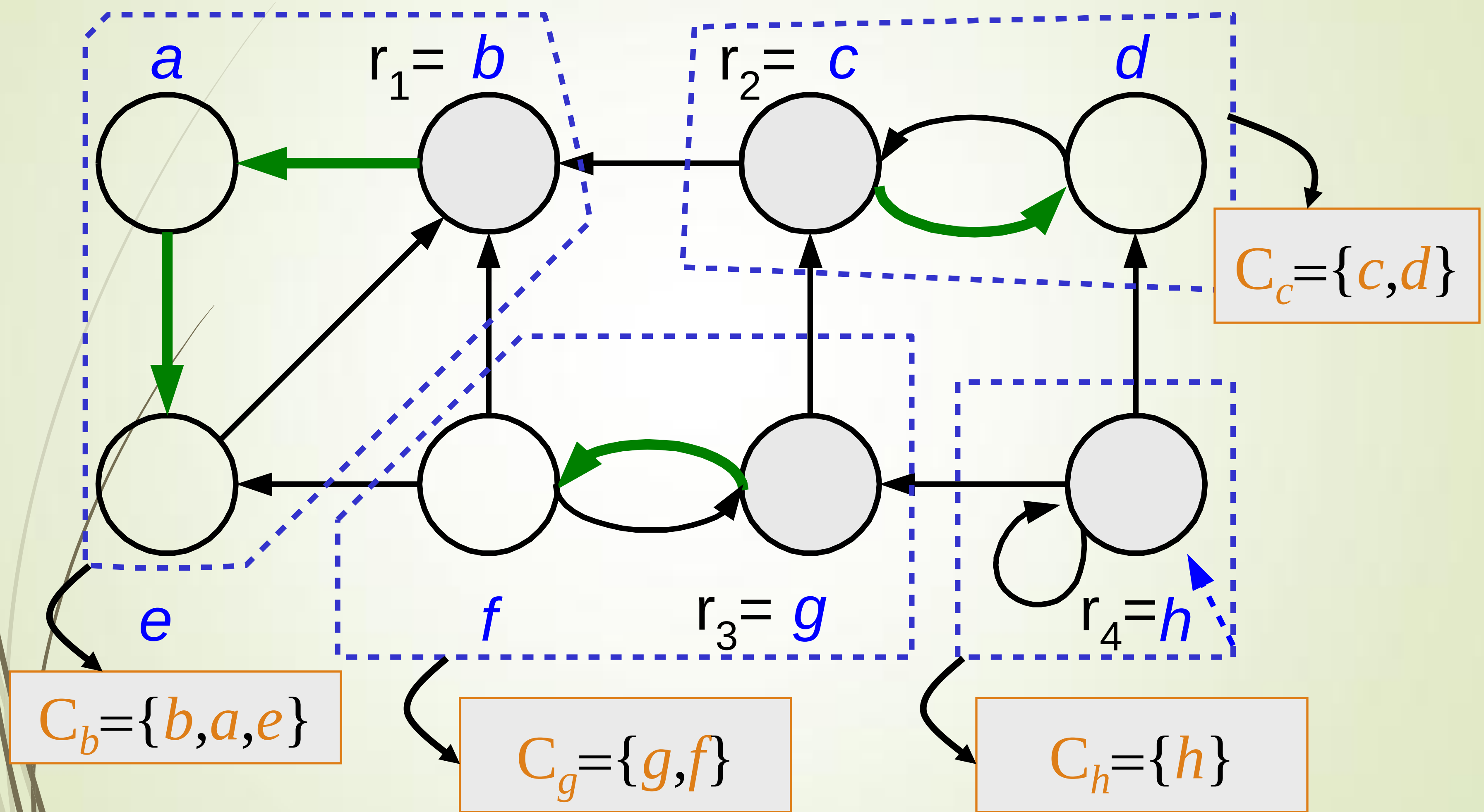
(3) Call **DFS**(G^T) processing vertices in main loop in decreasing $f[u]$ order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩

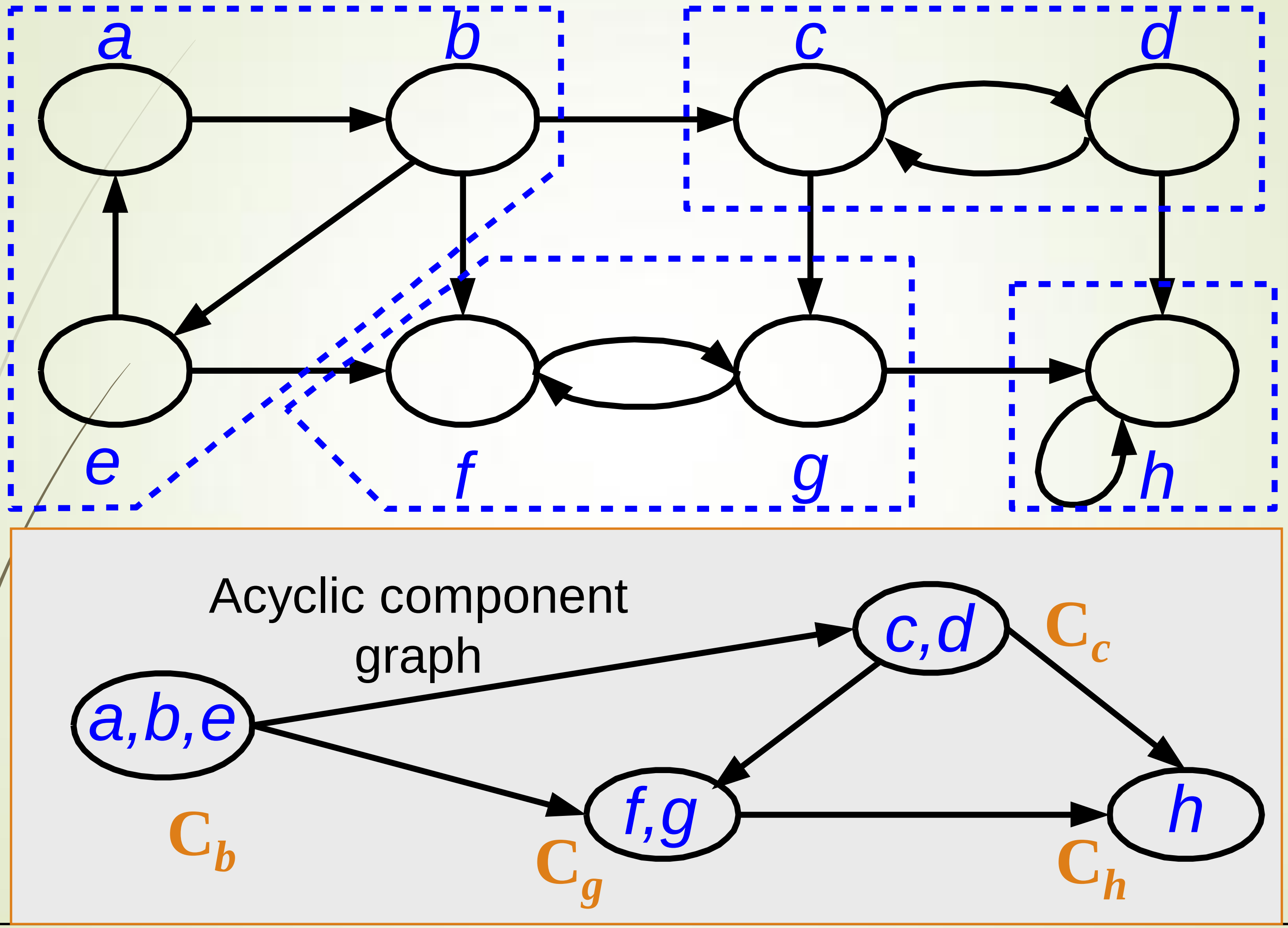


(3) Call DFS(G^T) processing vertices in main loop in decreasing f[*u*] order: ⟨***b***, ***e***, ***a***, ***c***, ***d***, ***g***, ***h***, ***f***⟩

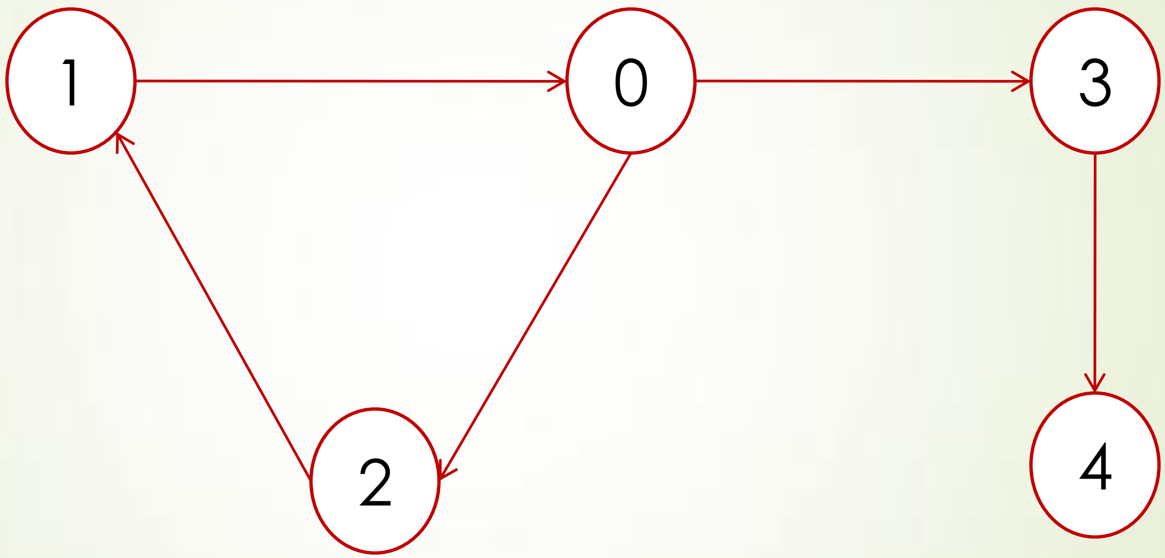


(4) Output vertices of each DFT in DFF as a separate SCC

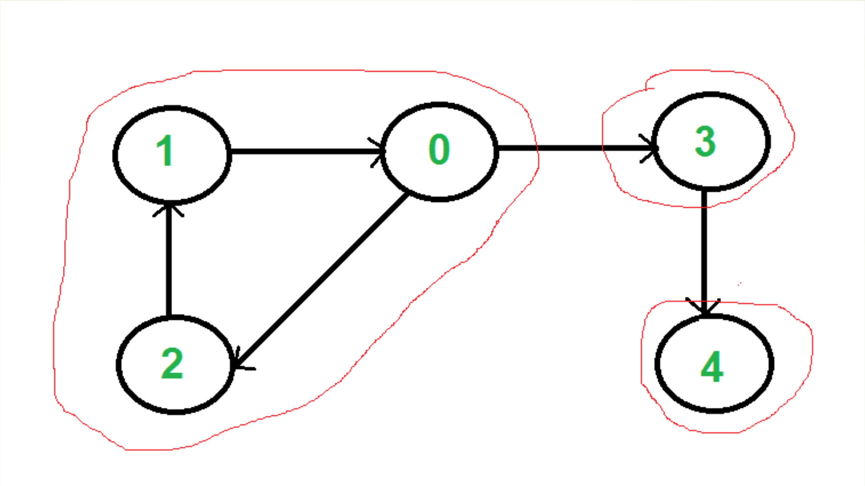


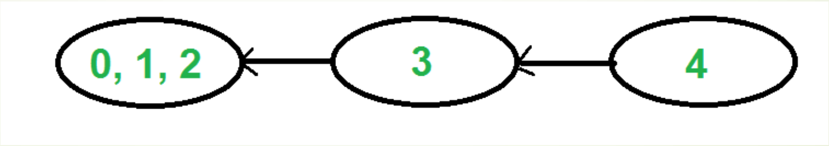

a
b
c
d
e
f
g
h
Acyclic component graph
a,b,e
c,d
f,g
h
C_b
C_c
C_g
C_h

EXAMPLE



EXAMPLE





END