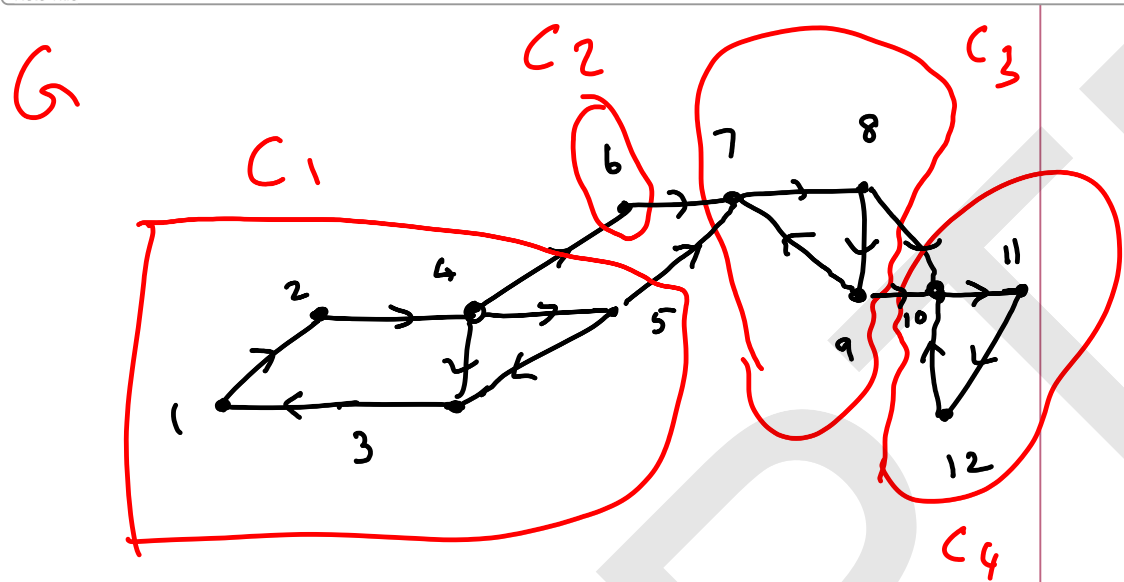
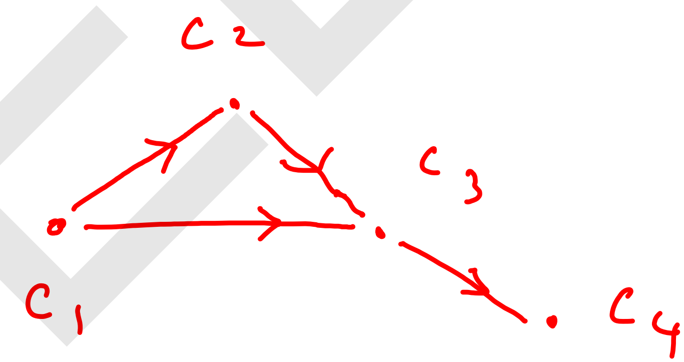
Note Title





Condensed graph of G.

→ DAG

" All vertices in a cycle will be in the same component "

$\underbrace{\{1,2,4,5,3\}}_{1}, \underbrace{\{6\}}_{2}, \underbrace{\{7,8,9\}}_{3}, \underbrace{\{10,11,12\}}_{4}.$

$SCC(2) = 1, \quad SCC(7) = 3$

if $scc(i) = j$.

$\Rightarrow$ vertex i is in the connected component j.

$V = \{1, 2, 3, \cdots n\}$.

Component numbers are $1, 2, 3, \cdots$

Input $= G = (V, E)$

output $= scc[\,]$

1) on G perform a DFS

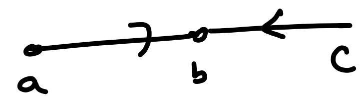
for each $v \in V$

$v.d$ – discovery time

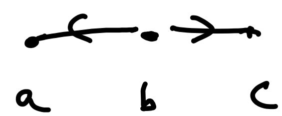
$v.f$ – finish time.

W $\quad v.d \quad$ G $\quad v.f \quad$ B.

2) Construct G^R.



G



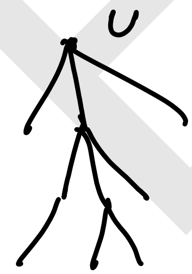
G^R.

③ In G^R perform DFS in the Decreasing order of $v.f$.

(Start every fresh search from an unvisited vertex with highest ($v.f$) finish time as done in step 1.

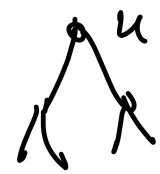
④ Output the vertices of each tree obtained in step 3 as component vertices.

Start DFS from a vertex v.
This stops after visiting all vertices reachable from v.



This will be a tree rooted at v.

If this tree does not contain all vertices of G, we continue the DFS from another unvisited vertex say u



$\longrightarrow$ tree rooted at u.

In general DFS will result in several trees, each rooted at a vertex and consisting of all vertices reachable from the root.

That is why, the "tree edges" $(p(v), v)$ form a forest in general.

(several trees).

The tree vertices form a partition of the vertex set V.

"DFS forest"

We note that the components induced by DFS ~~for~~ on G^R are the strong connected components of G.