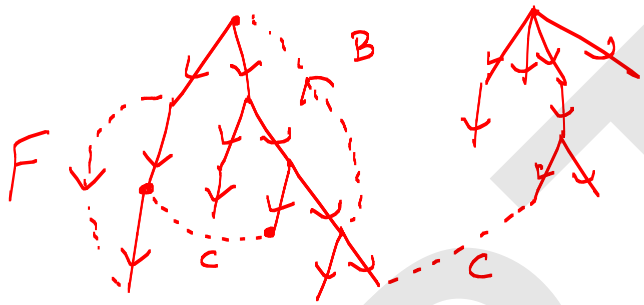
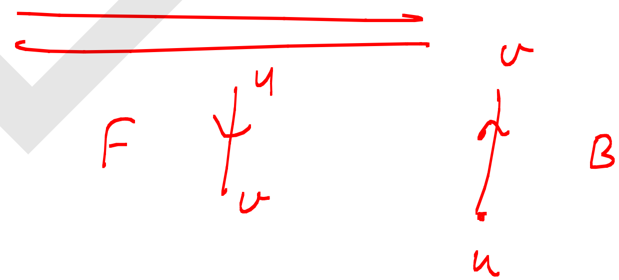
DFS in Directed Graph.

Note Title



$e = (u, v)$



Visited node $\longrightarrow$ unvisited node — Tree edge.

Visited node $\longrightarrow$ visited node — Non Tree edges.

Non Tree edges

- F — Ancestor-to-Desc.
- B — Desc to Anc
- C — Cross.

Time — 1 to 2n consecutive integers.

u — [u.d, u.f]

1 ——— 2n

active

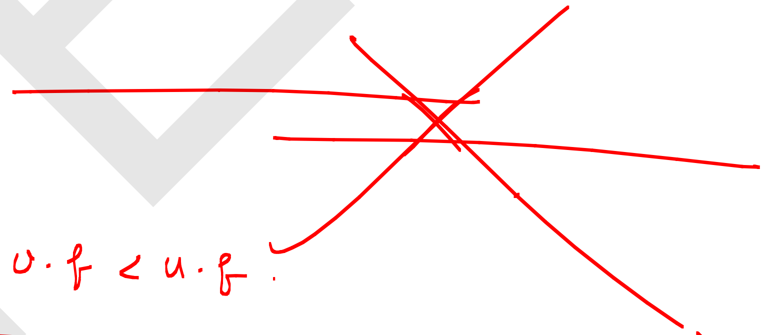
W	G	B
Unvisited	Active	Finished

↑ u.d   ↑ u.f

u v

$u.d$ u $u.f$

$v.d$ v $v.f$



$u.d < v.d < v.f < u.f$

v is desc of u.

DFS (G)

For each $u \in V$

$u.col = white$

$u.p = NULL.$

$Time = 0$

For each $u \in V$

if ($u.col == white$)

DFS (G, u).

DFS (G, u)

$Time = Time + 1;$

$u.d = Time;$

$u.col = Gray;$

For each $v \in Adj[u]$

if ($v.col == white$)

$v.p = u;$

\\ (u, v) is a Tree edge .

DFS (G, v).

else \\ (u, v) is a Non-Tree edge .

else // (u, v) is a non tree edge.

elaboration of Control point related to Non-Tree edges.

if (v.col == Gray)

// (u, v) is a back edge.

if (v.f < u.d)

// (u, v) is a cross edge.

else

// Forward edge.

Finding strongly connected Components of a Directed Graph.