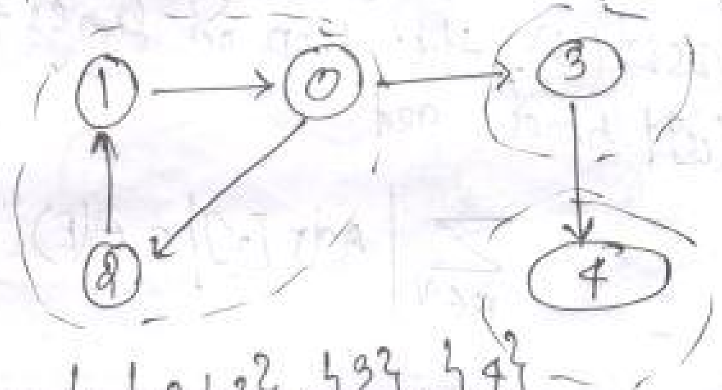
Strongly connected component

↳ A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component (SCC) of a directed graph is a maximal strongly connected subgraph. For example there are 3 SCC, in the following gr



SCC = { {0,1,2}, {3}, {4} }

⇒ we can find all strongly connected components in O(V+E) time using Kosaraju's algorithm.

KOSARAJU ALGORITHM

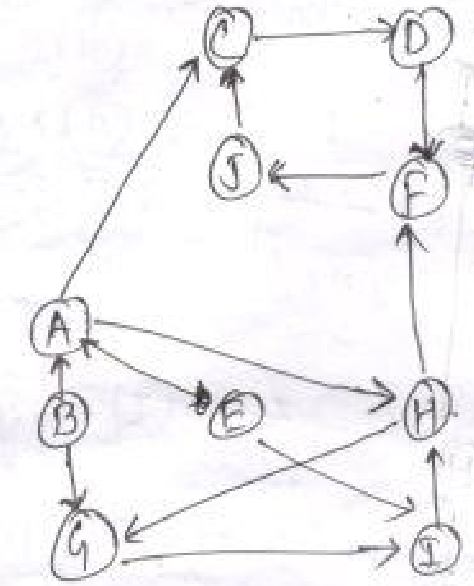
① Call DFS(G) to compute finishing time f[v] for each vertex u.

② Compute Transpose (G).

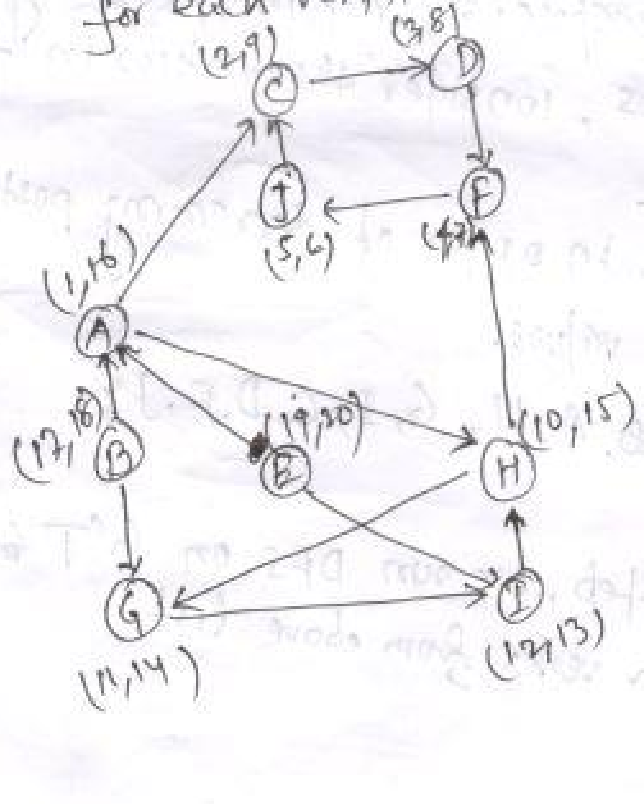
③ Call DFS(Transpose(G)), but in the main loop of consider the vertices in order of decreasing f

④ Output the vertices of each tree in depth-f forest of step 3 as a seprate strong connected c

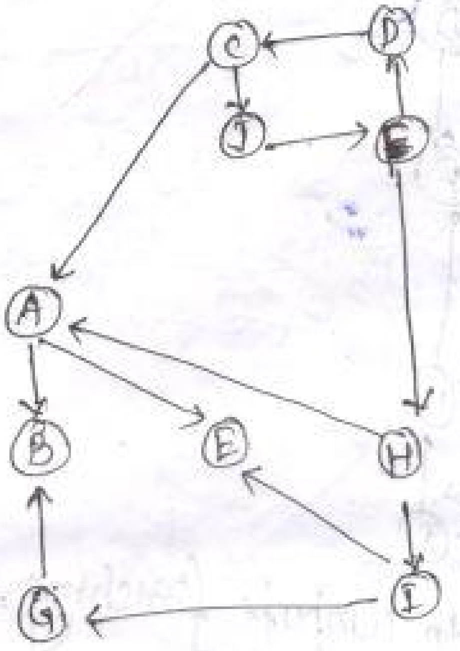
Ex



Step I Call DFS(G) to compute finishing times $f[u]$ for each vertex u.



Step2 Compute the transpose (G)



Step3 Call DFS (transpose(G)), but in ~~decreasing~~ the main loop of DFS, consider the vertices in dec order of f[u].

→ Okay, so vertices in order of decreasing post-va (finishing times) values.

{E, B, A, H, G, I, C, D, F, J}

→ So at this step, we run DFS on G^T but start with each vertex from above list.

DFS(E) = {E}

DFS(B) = {B}

DFS(A) = {A}

DFS(H) = {H, I, G}

DFS(G) = {remove from list since it is already visited

DFS(I) = "

DFS(C) = {C, J, F, D}

DFS(J)
DFS(F)
DFS(D)
} removed from list as it is already visited.