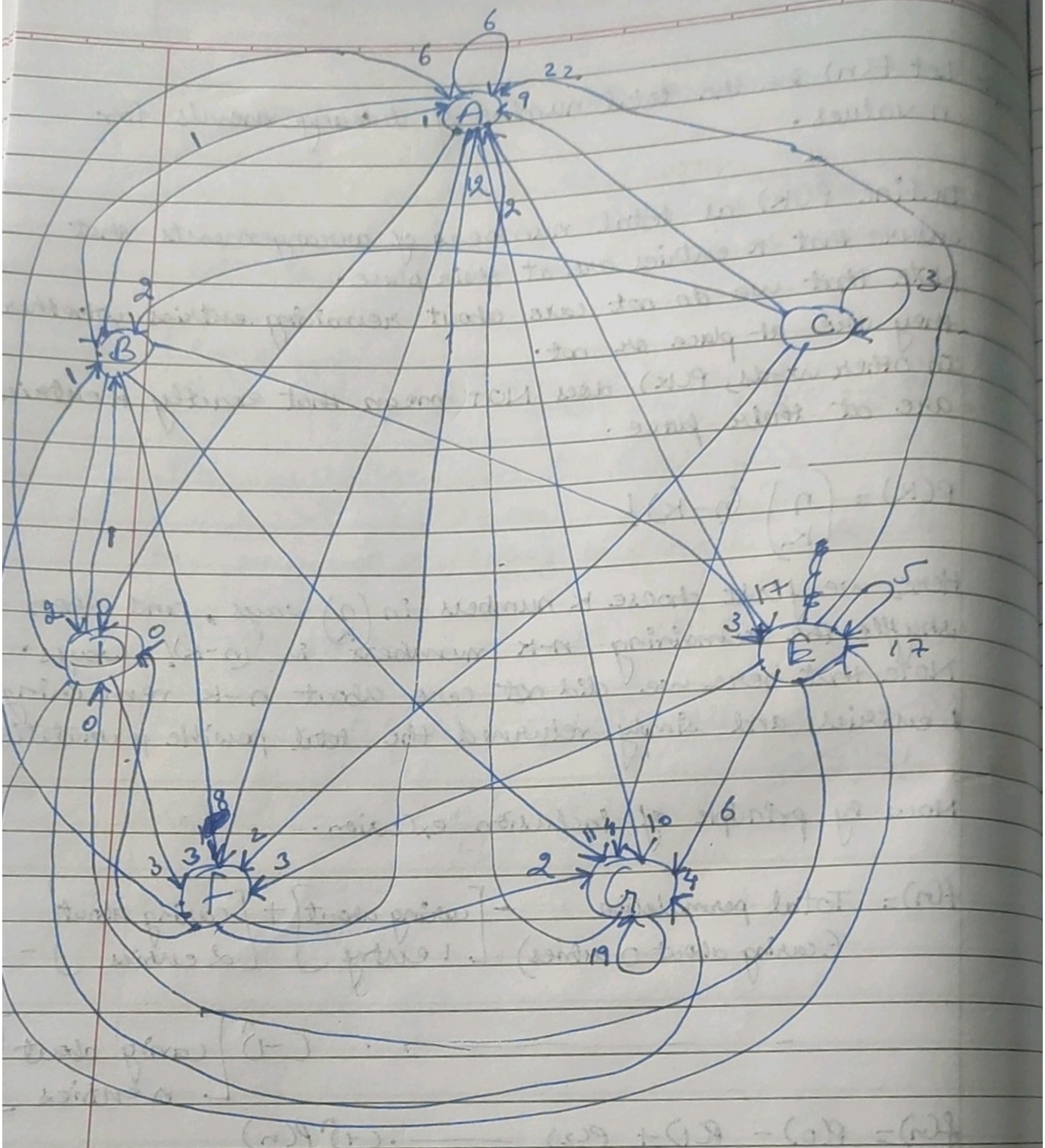
Deepanshu 2019CS50427

Page:
Date:



We assumed that the entry at i^{th} row and j^{th} column means that there is an edge from i^{th} vertex to j^{th} vertex having the weight equal to the value at i^{th} row and j^{th} column in the matrix representation of the graph.

Deepanshu 2019CS50427

Page :
Date :

Adjacency list representation

(A) 1 → [1 | 6 |] → [2 | 1 |] → [4 | 0 |] → [5 | 17 |] → [6 | 3 |] → [7 | 10 |]
(B) 2 → [1 | 1 |] → [4 | 2 |] → [5 | 3 |] → [6 | 8 |] → [7 | 11 |]
(C) 3 → [1 | 9 |] → [2 | 2 |] → [3 | 3 |] → [6 | 2 |] → [7 | 4 |]
(D) 4 → [1 | 6 |] → [2 | 1 |] → [5 | 17 |] → [6 | 3 |] → [7 | 4 |]
(E) 5 → [1 | 22 |] → [4 | 0 |] → [5 | 5 |] → [6 | 3 |] → [7 | 6 |]
(F) 6 → [1 | 12 |] → [2 | 1 |] → [4 | 0 |] → [7 | 2 |]
(G) 7 → [1 | 2 |] → [7 | 19 |]

Homework 22-2

Ans ⇒ order of visiting vertices:

~~A → D → C~~

A → B → D → E → F → G → G → F → E → D → B → A → C → C

Arrival time	Departure time
A : 1	A : 12
B : 2	B : 11
C : 13	C : 14
D : 3	D : 10
E : 4	E : 9
F : 5	F : 8
G : 6	G : 7

Tree Edges: A → B, B → D, D → E, E → G, G → H

Cross Edges: C → A, C → B, ~~C → F~~, C → G

forward edge: A → B, A → D, A → E, A → F, A → G, B → D, B → E, B → G, B → G, D → E, D → F, D → G, E → F, E → G, F → G

Back edge: B → A, C → B, C → A, C → F, C → G, D → B, D → A, E → A, E → D, F → A, F → B, F → D, G → A

Deepanshu 2019CS50427

Page :
Date :

Homework 22.3

We are given two DAG's $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$

For the supergraph to have cycle, there should be a back edge.
Suppose (u,v) is a back edge.

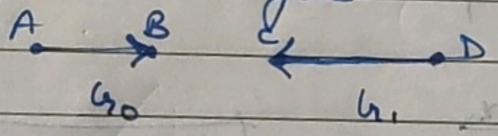
Now while performing topological sort on supergraph, since (u,v) is edge from vertex in G_0 to vertex in G_1, all vertices of G_0 would appear before u i.e before departure time of u.

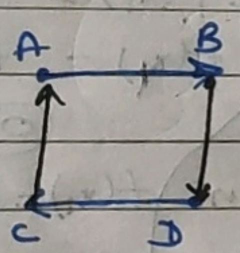
Also, now $arr[v] > arr[u]$ and $dep[v] < dep[u]$
But these are properties of forward edge.

$\Rightarrow$ Inserting one edge connecting two DAG's would make it a forward edge of supergraph.
~~supergr~~ supergraph is still a DAG.

If another such edge is added, the supergraph may become cyclic

Eg.





Supergraph with cycle after adding 2 edges.

Therefore, after adding 2 such edges, the resulting supergraph may have a cycle.

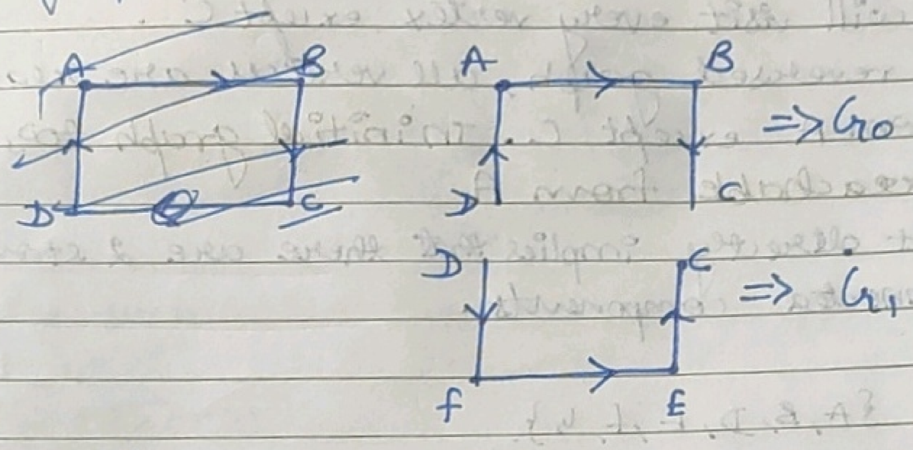
Deepanshu 2019CS50427

Observation:

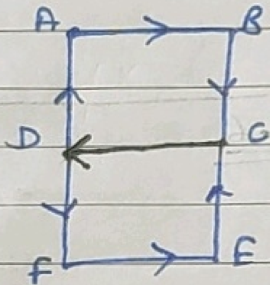
If we relax our assumption a bit and assume that $V_0 \cap V_1 \neq \emptyset$

Then one edge can also ~~not~~ generate a cycle in supergraph

eg.



Supergraph:



It has a cycle.
In fact, it is also strongly connected.

Homework 22.4

The resulting ordering would be such that every edge (u, v) u will come before v.

Whereas in decreasing arrival time ordering, for every edge (u, v) v will come before u.

Deepanshu 2019CS50427

Page :
Date :

Homework 22.5

Ans=> If we follow the algo specified by Dr. Naveen in which we ~~po~~ first perform DFS, reverse the directed graph and again perform DFS.

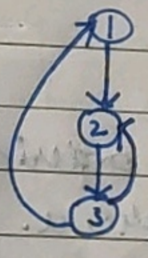
It will visit every vertex except C.
In reversed graph, all vertices are reachable from A except C. In initial graph too, c is not reachable from A.
~~E~~ It directly implies that there are 2 strongly connected components.

1 : {A, B, D, E, f, G}
2: {C}.

Homework 22.6.

for lower bound
After removal of an edge, it cannot happen that one vertex become now accessible (after removing).
Thus, number of strongly connected components cannot decrease.

eg.



Removing edge from 3 to 2 would have no change on number of strongly connected components.

Deepanshu 2019CS50427

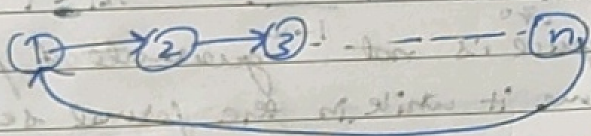
Page :
Date :

for upper bound

For upper bound, removal of an edge can make the count of strongly connected components rise up to the total number of vertices.

It happens in cases when there is a long chain of vertices connected ~~to it~~ in one direction. And an edge from tail of ~~edge to head~~ chain to head of chain making all vertices accessible.

Removing such an edge would be the case for upper bound



Removing the edge from n to 1 would make the count of strongly connected components to rise from 1 to n.