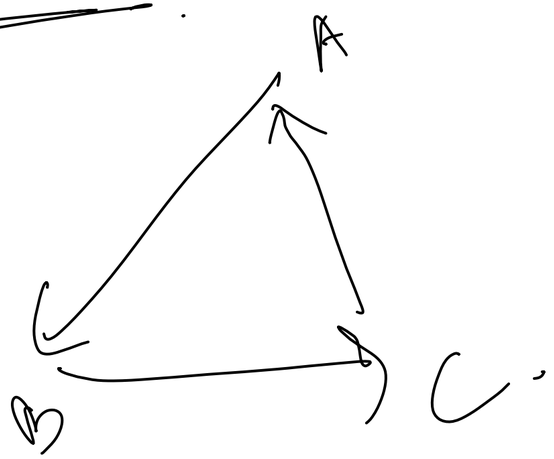
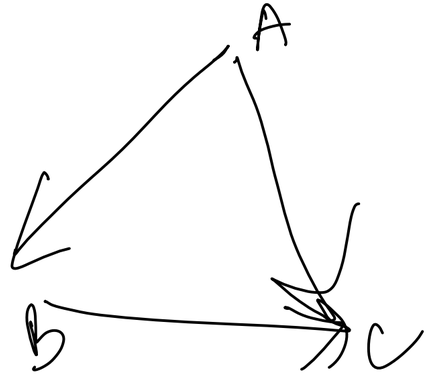
4.11.2024

Directed Acyclic Graphs (DAG): Directed graphs which does it have a cycle!

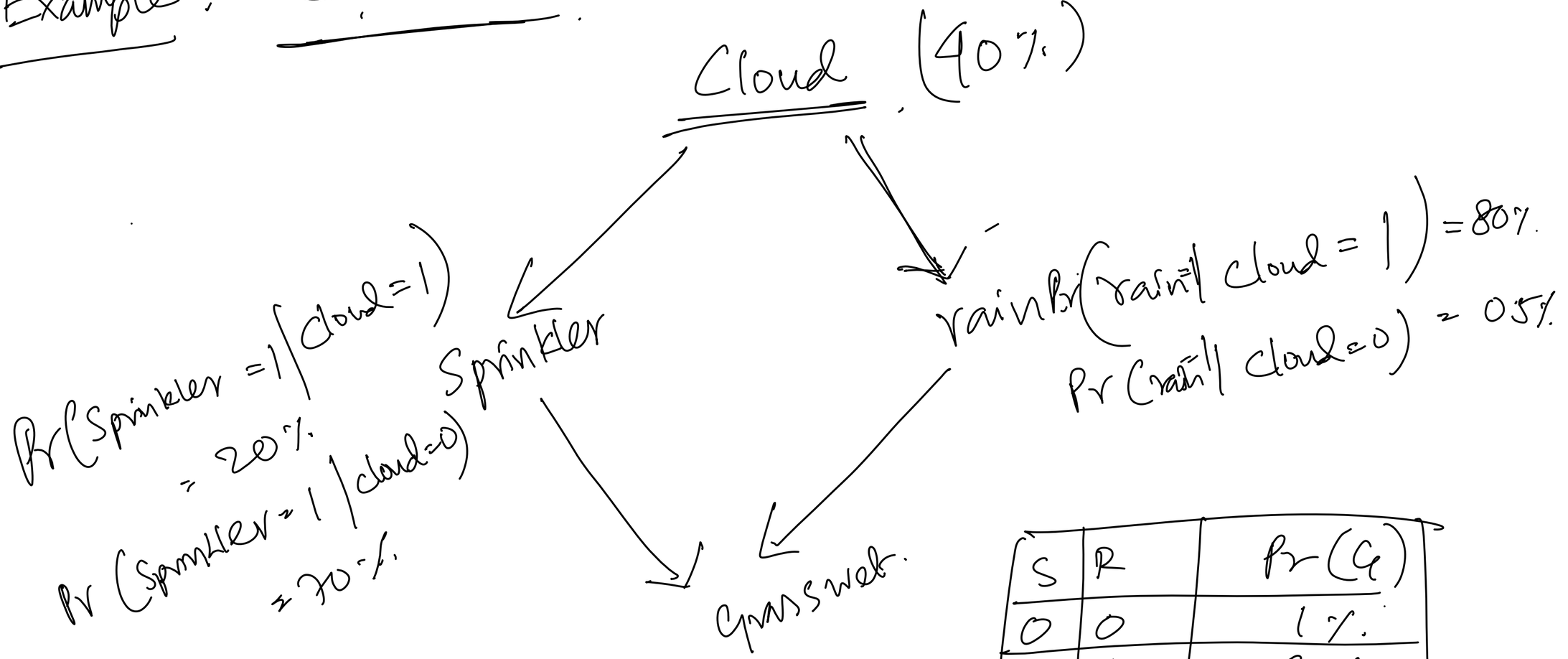


not



DAG

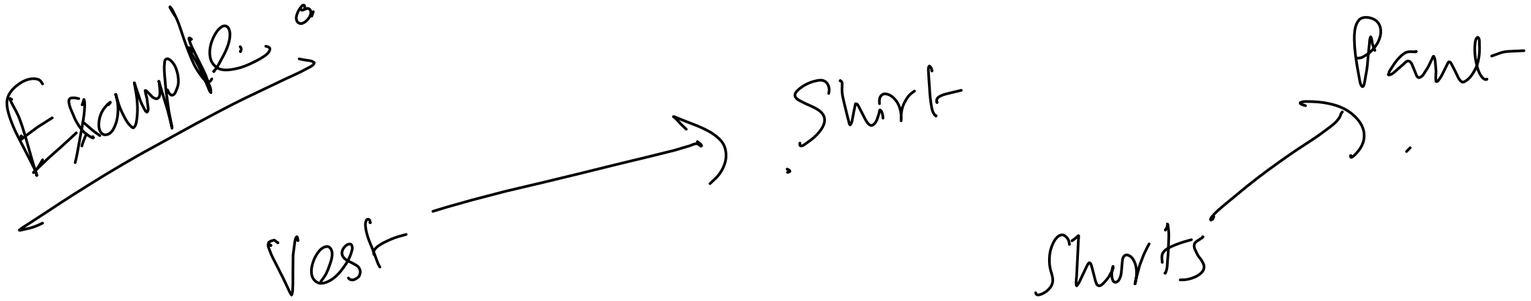
Example: Causal:



S	R	Pr(G)
0	0	1%.
0	1	90%.
1	0	90%.
1	1	99%.

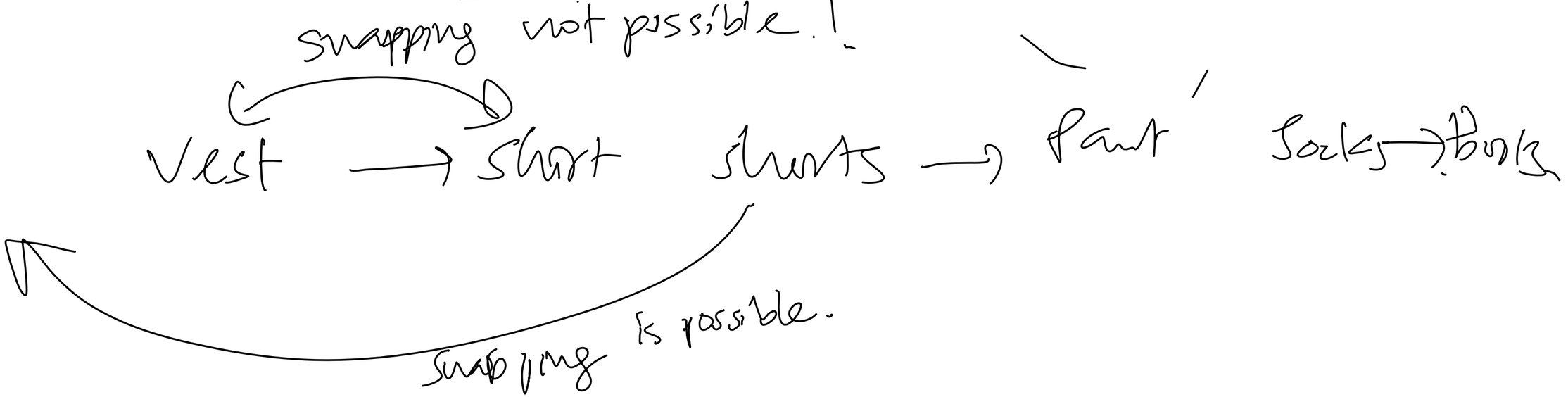
DAG.

Example:

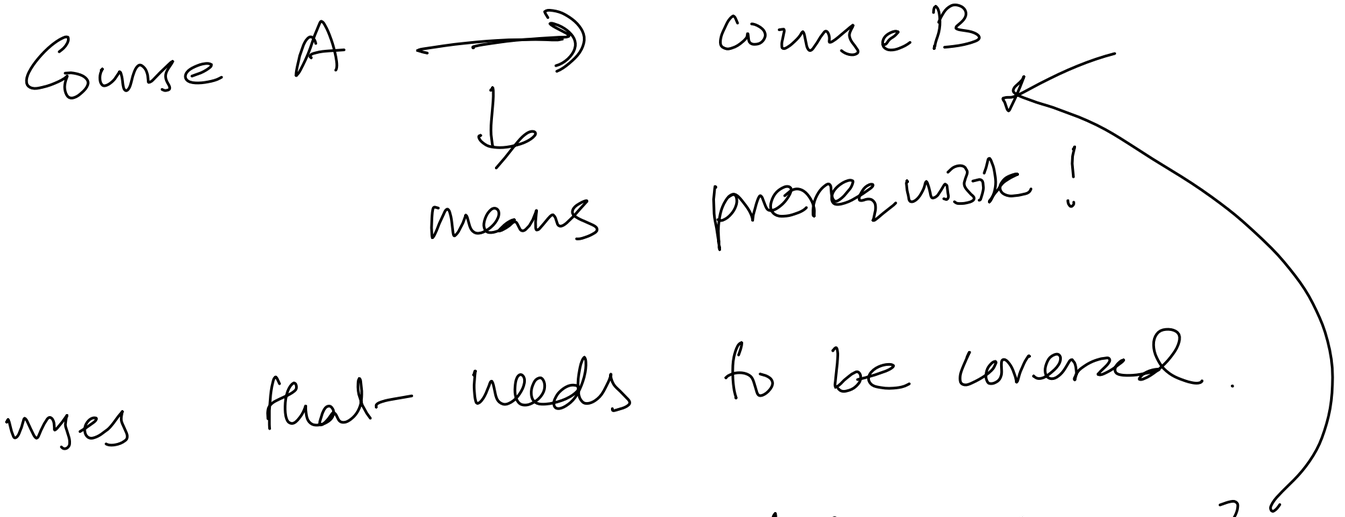


Partial Order

DAG



Example 3

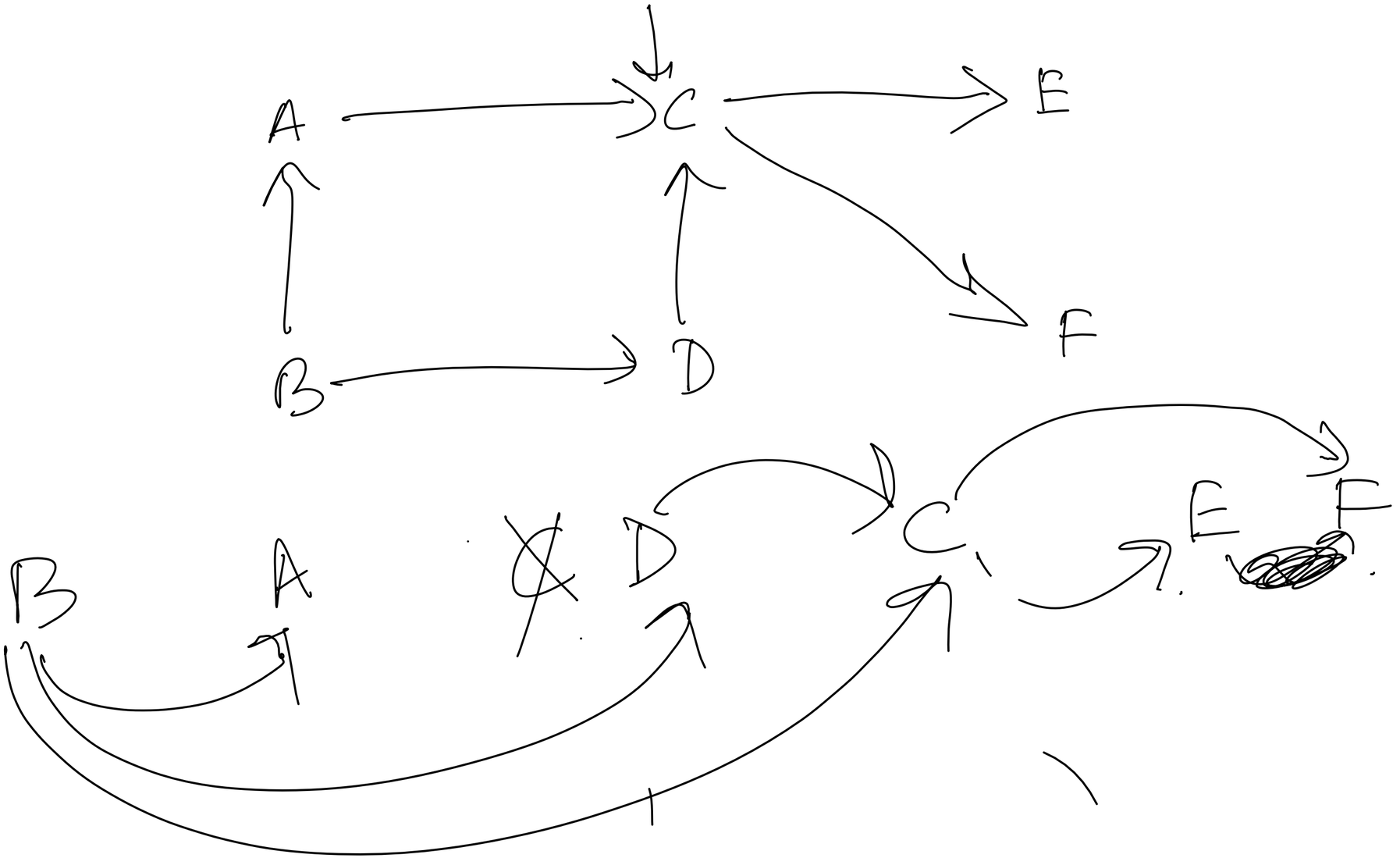


A set of courses that needs to be covered.

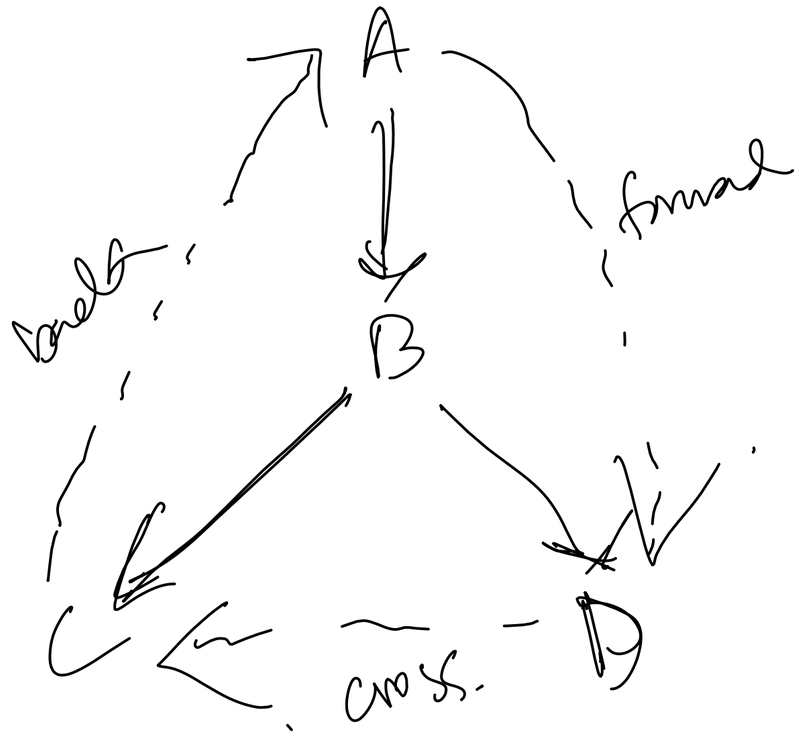
Can we do ~~an~~ ~~cde~~ a valid ordering?

— Yes!

Given a DAG we can always sort it in a manner so that edges always go left to right! (Topological sort/ Topo. sort)



edges go from left to right!



→ tree edges.

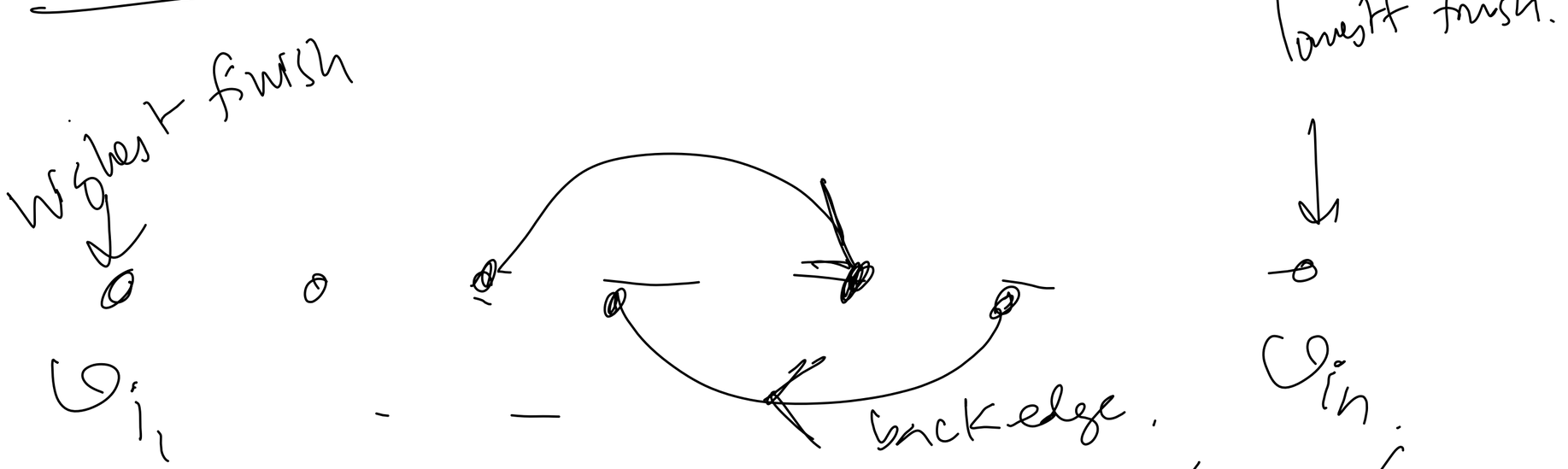
(Recall DFS)

~~Source node~~

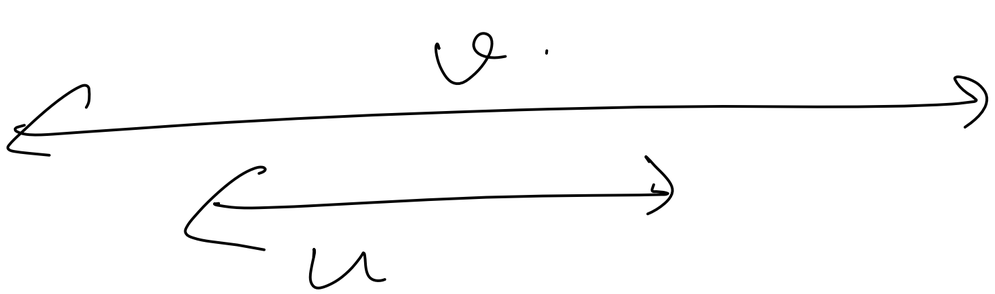
Fact: The vertex having highest finish time will always be a source node.

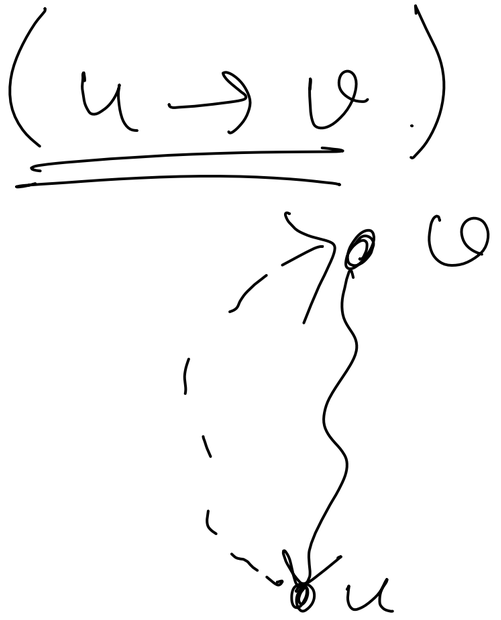
Fact: If we sort the vertices from highest finish time to low, that will be a valid topological sort.

pf: ←



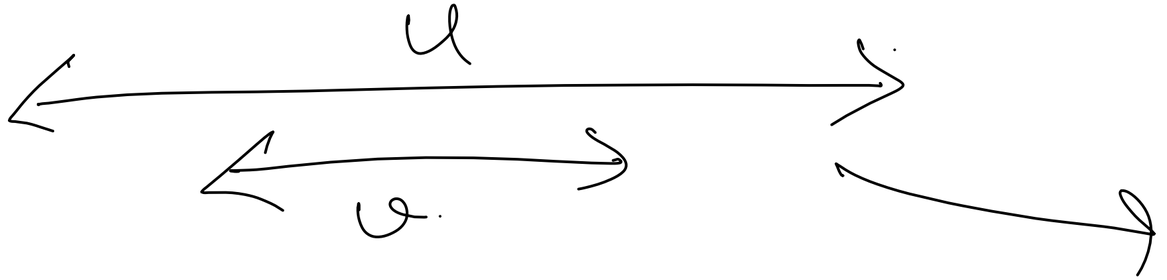
what is wrong if an edge goes from low to high finish time?

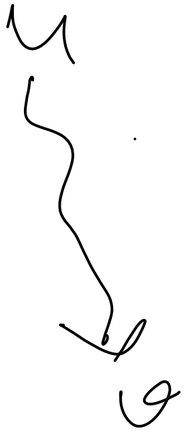




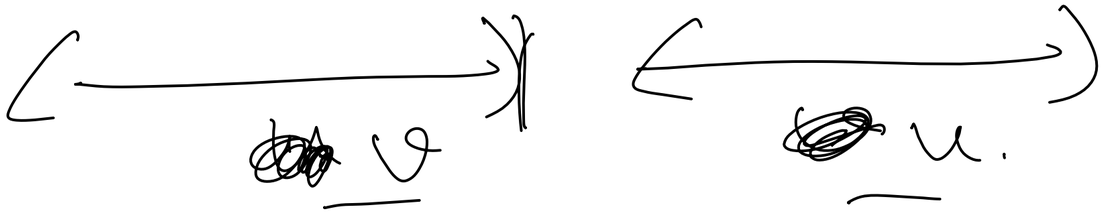
Only happens for back edge

tree / forward





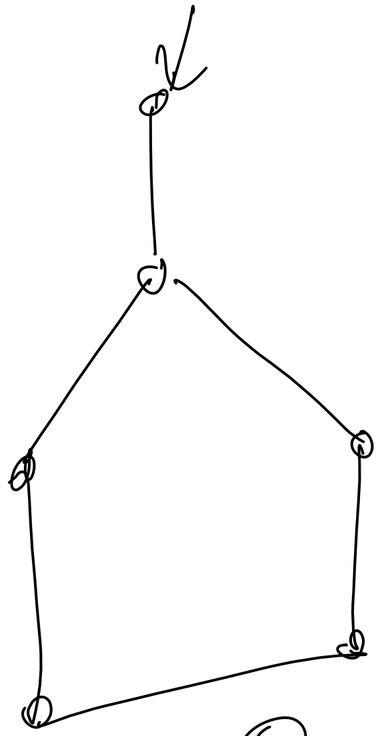
cross edge



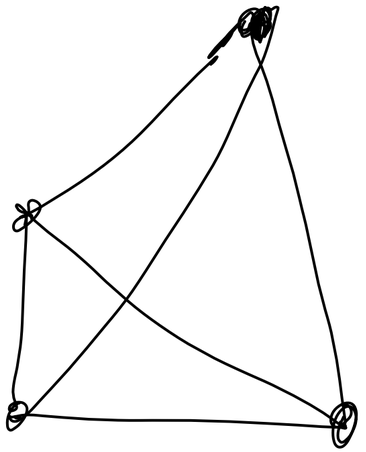
v's finish time is smaller than u.

We know there are no cycle $\Rightarrow$ no back edges.

So the above is a valid topo. sort! □.



0

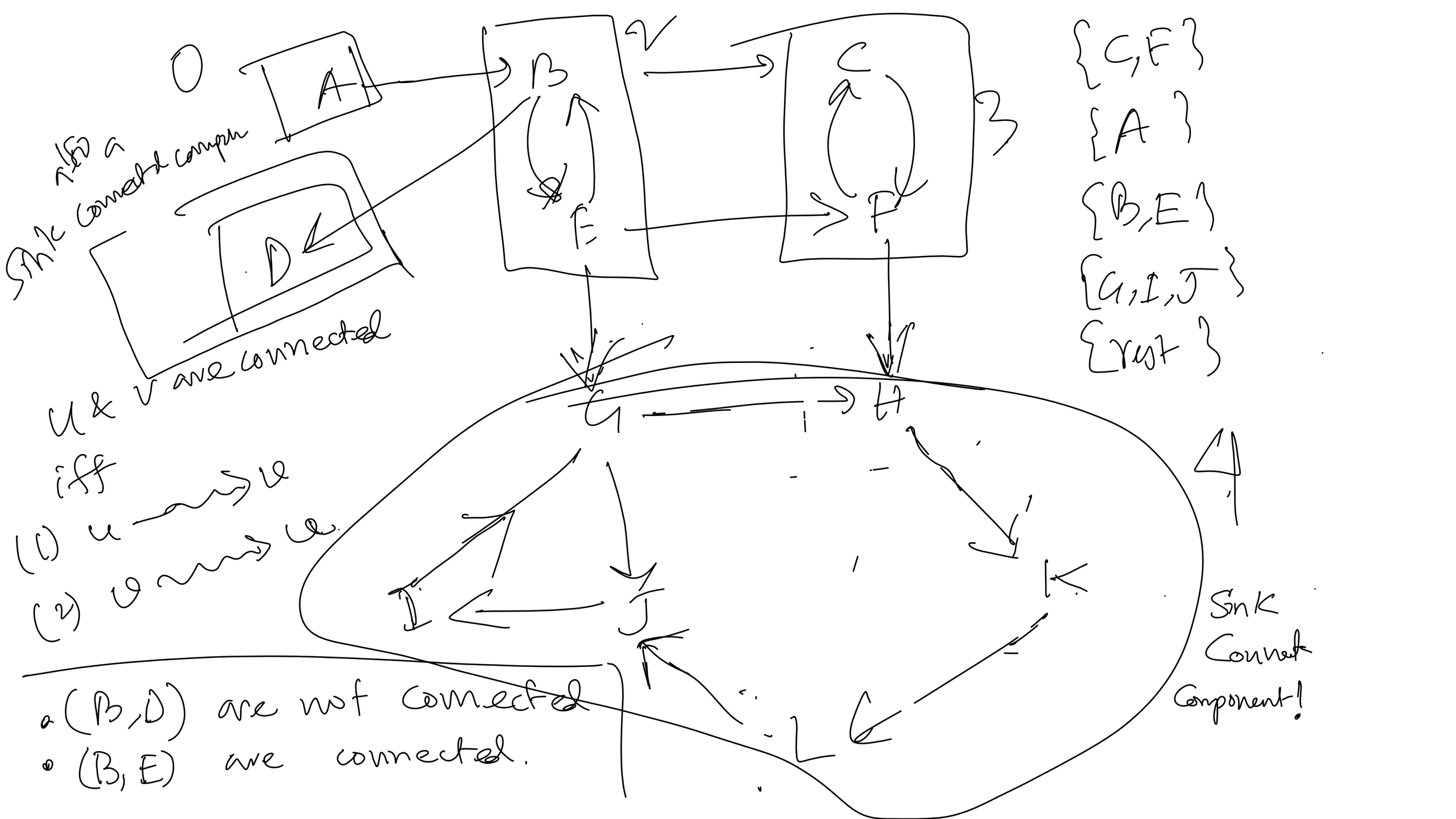


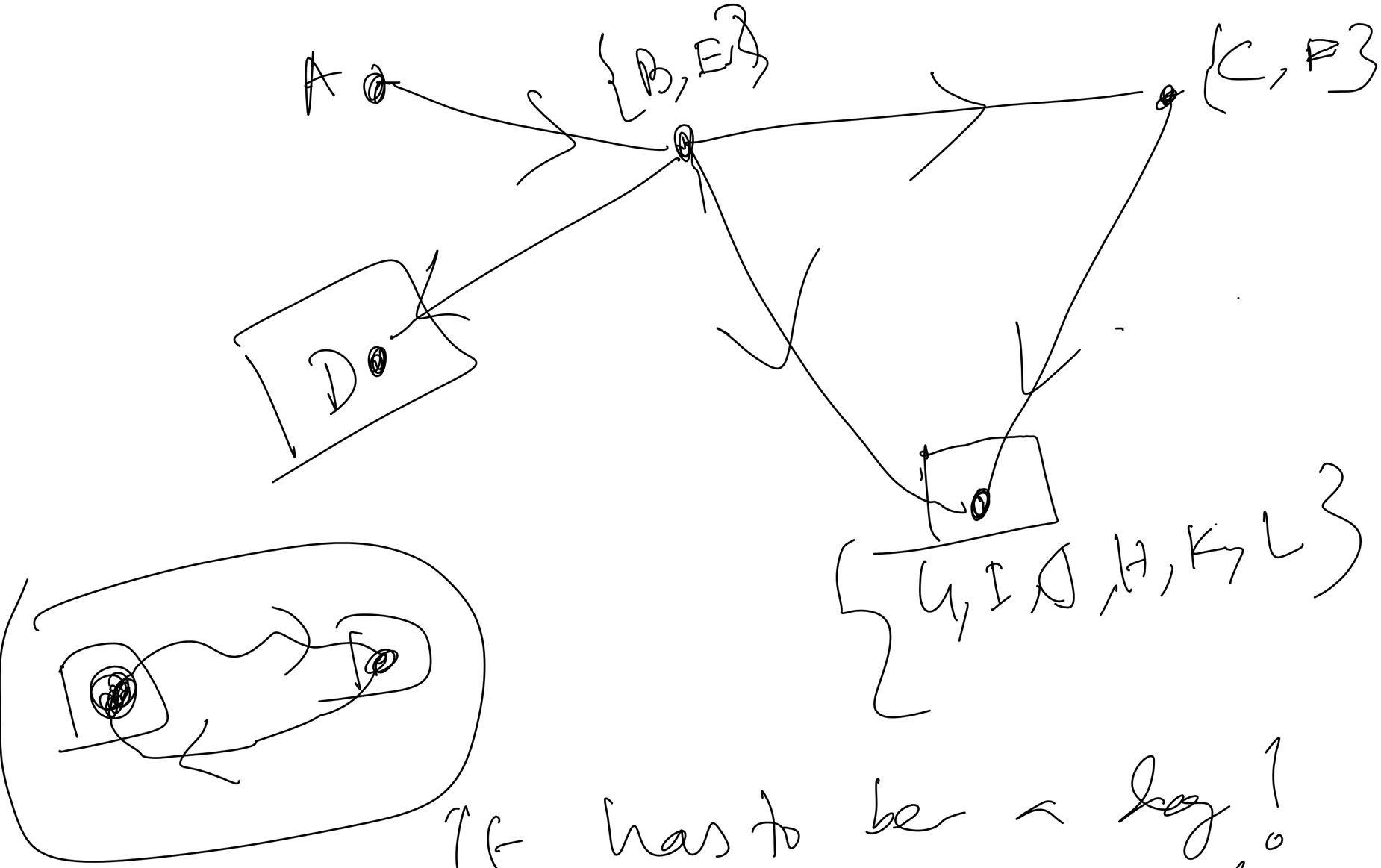
1

10 nodes.

tot # components = 2.

$O(|V|+|E|)$


A
B
C
D
E
F
G
H
I
J
K
L
{C,F}
{A}
{B,E}
{G,I,J}
{rest}
u & v are connected
iff
(1) u ⇝ v
(2) v ⇝ u
Sink Connect Component!
• (B,D) are not connected
• (B,E) are connected.



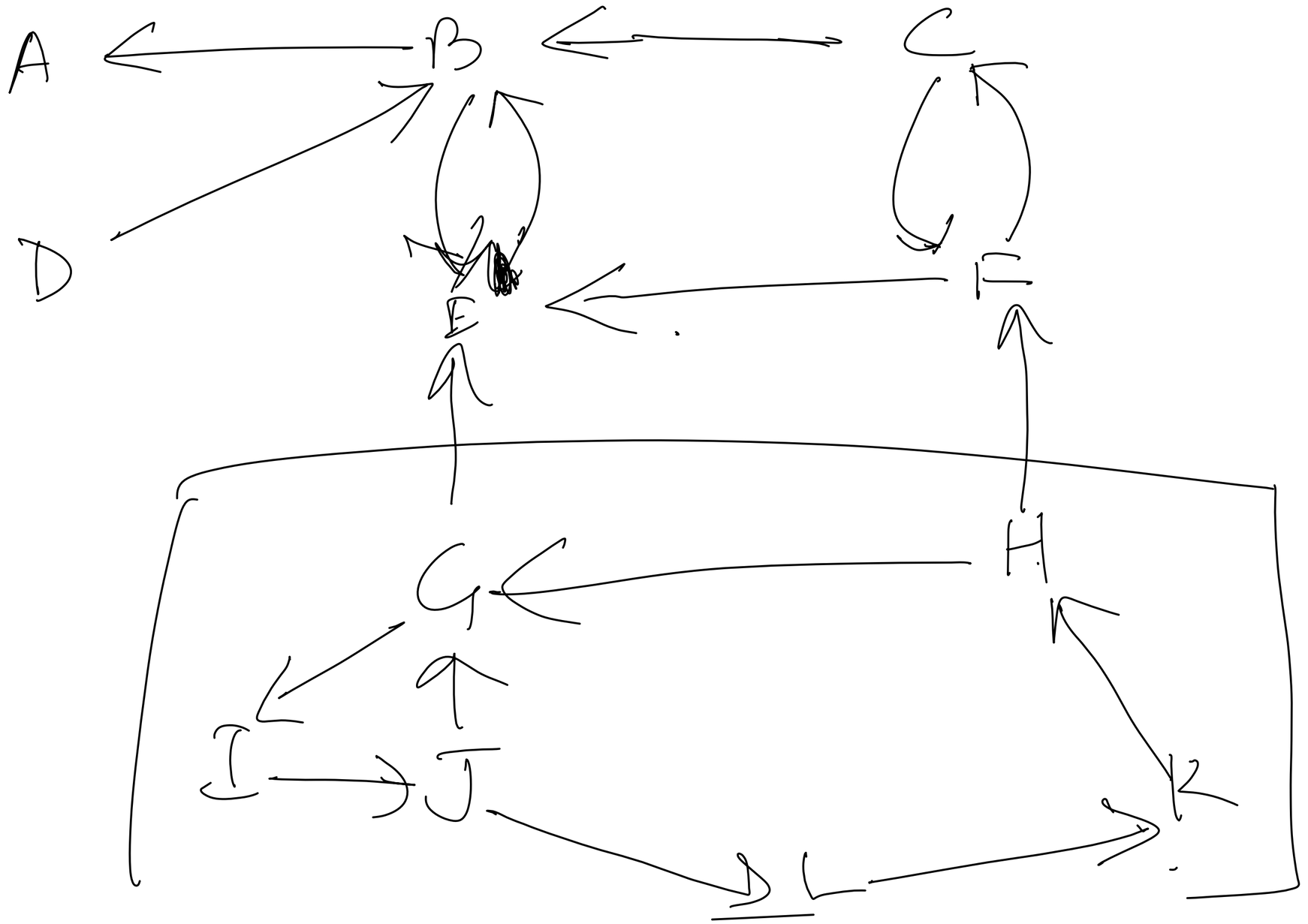
If has to be ~ log!

≡ no cycles here!

Fact: It's easy to find a source node using DFS but not a sink node.

Corollary: If we reverse the graph, the last finish time will be part of a sink connected component.

reverse of previous graph



(1). Reverse the given graph G to get a graph G^R.

(2). perform DFS on G^R & sort from high to low finish time.

or G.

(3) Start DFS according to above sorting order.