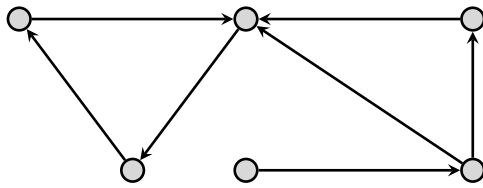


## 1 Strongly Connected Components

Consider the directed graph below:



How many strongly connected components does this graph have?

Correct

What is the minimum number of directed edges to add to this graph to make all the vertices strongly connected?

Correct

Assume you have two vertices  $u$  and  $v$  in a directed graph where there exists a path from  $u$  to  $v$ . Which one of the following is incorrect about  $u$  and  $v$ ?

- $u$  and  $v$  can be in the same SCC.
- $u$  and  $v$  can be in different SCCs.
- If  $u$ 's DFS finish time is less than  $v$ 's DFS finish time then  $u$  and  $v$  are in the same SCC.
- $u$ 's DFS finish time is always greater than  $v$ 's DFS finish time.

Correct

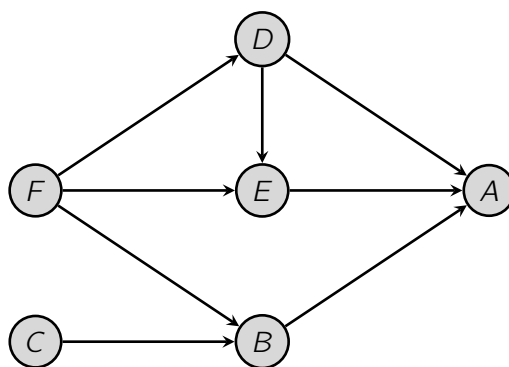
Assume you have two vertices  $u$  and  $v$  in a directed graph where  $u$  and  $v$  are in the same SCC. Which one of the following is incorrect about  $u$  and  $v$ ?

- There exists a DFS tree where  $u$  is in  $v$ 's subtree (subtree rooted at  $v$ ).
- There exists a DFS tree where  $v$  is in  $u$ 's subtree (subtree rooted at  $u$ ).
- There exists a DFS tree where  $u$  is not in  $v$ 's subtree and  $v$  is not in  $u$ 's subtree.
- There exists a BFS tree where  $u$  is not in  $v$ 's subtree and  $v$  is not in  $u$ 's subtree

Correct

## 2 Topological Sorting

Consider the DAG below:



How many different orderings of the vertices in the above graph (out of the  $6!$  possible orderings) result in a topological sort? For instance  $ABCDEF$  is an ordering that's not topologically sorted, but  $FDCEBA$  is an ordering that's topologically sorted.

Correct

What is the lexicographically smallest topological ordering of the vertices?

Correct

If we use a DFS algorithm that breaks ties lexicographically (always picks the node with lexicographically smallest letter possible to start or proceed), what is the resulting topological ordering of the nodes?

Correct