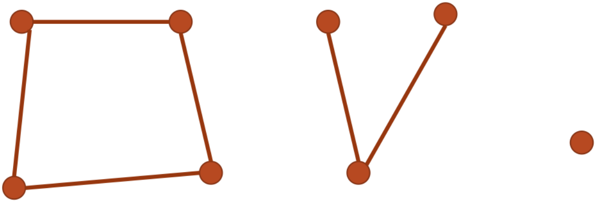
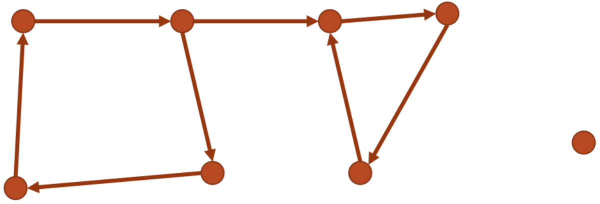
COL106: Data Structures and Algorithms

Ragesh Jaiswal, IIT Delhi

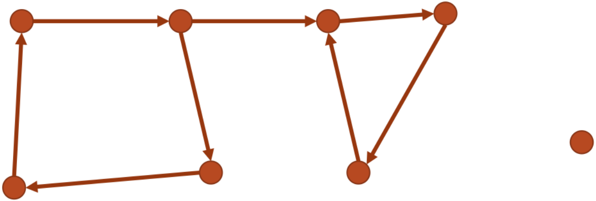
- A graph may not always be "connected".
- A connected component in an undirected graph is a maximal subgraph (maximal subset of vertices along with respective edges) such that there is a path between any pair of vertices in the subset.



- In a directed graph, a strongly connected component is a maximal subgraph such that for each pair of vertices (u, v) in the subset, there is a path from u to v and there is a path from v to u.



- Question: Given a directed graph, can a vertex be in two strongly connected components?

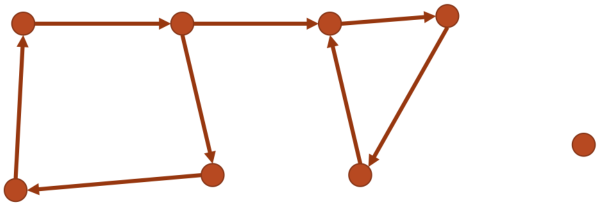


- Question: Given a directed graph, can a vertex be in two strongly connected components? No
  - For sake of contradiction, assume that there is a vertex v and vertex sets A, B in two strongly connected components s.t. $v \in A$, $v \in B$ and $A \neq B$.
  - Claim: For ever pair of vertices $p, q \in A \cup B$, there is a path from p to q and there is a path from q to p.
  - This implies that either A or B is not a *maximal* subset.

- Question: Given a directed graph, can a vertex be in two strongly connected components? No

Problem

Given a directed graph and a vertex s. Give an algorithm to find the vertices in the strongly connected component containing s. What is the running time?

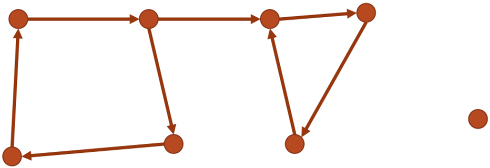


Problem

Given a directed graph and a vertex s. Give an algorithm to find the vertices in the strongly connected component containing s. What is the running time?

Algorithm

`SCC-containing-s`(G, s)
- Do $DFS(s)$ on G and let A be the vertices that are explored.
- Let G^R be the graph obtained by reversing the edges of G
- Do $DFS(s)$ on G^R and let B be the vertices that are explored.
- Output$(A \cap B)$



Problem

Given a directed graph and a vertex s. Give an algorithm to find the vertices in the strongly connected component containing s. What is the running time?

Algorithm

`SCC-containing-s`(G, s)

- Do $DFS(s)$ on G and let A be the vertices that are explored.
- Let G^R be the graph obtained by reversing the edges of G
- Do $DFS(s)$ on G^R and let B be the vertices that are explored.
- Output$(A \cap B)$

Proof (sketch) of correctness

- Claim 1: For every $u, v \in A \cap B$, there is a path in G from u to v and from v to u.

Problem

Given a directed graph and a vertex s. Give an algorithm to find the vertices in the strongly connected component containing s. What is the running time?

Algorithm

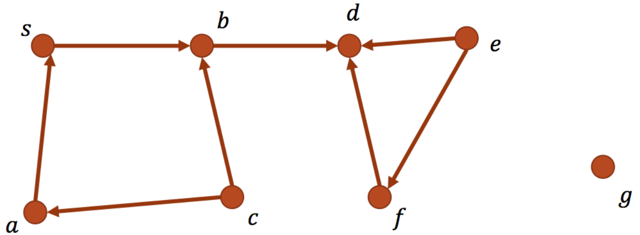
`SCC-containing-s`(G, s)

- Do $DFS(s)$ on G and let A be the vertices that are explored.
- Let G^R be the graph obtained by reversing the edges of G
- Do $DFS(s)$ on G^R and let B be the vertices that are explored.
- Output$(A \cap B)$

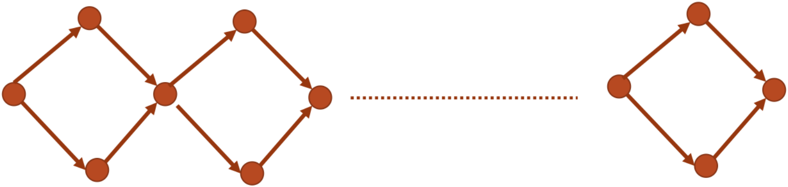
Proof (sketch) of correctness

- Claim 1: For every $u, v \in A \cap B$, there is a path in G from u to v and from v to u.
  - Both the paths go through s.
- Claim 2: $A \cap B$ is the maximal subset satisfying condition in Claim 1.

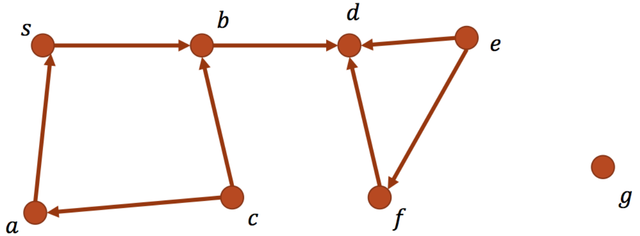
- Directed Acyclic Graph (DAG): A directed acyclic graph is a directed graph such that there are no cycles in the graph.
- Topological ordering: An ordering of the vertices of a directed graph such that there is no directed edge from a vertex that lies later in the order to another vertex that lies earlier in the order.



- Question: How many topological ordering of the following graph is possible?



- Question: Given a directed graph that contains a cycle. Is topological ordering possible?
- Question: Given a DAG. Is topological ordering possible? If so give an algorithm that outputs one such order. What is the running time?



End