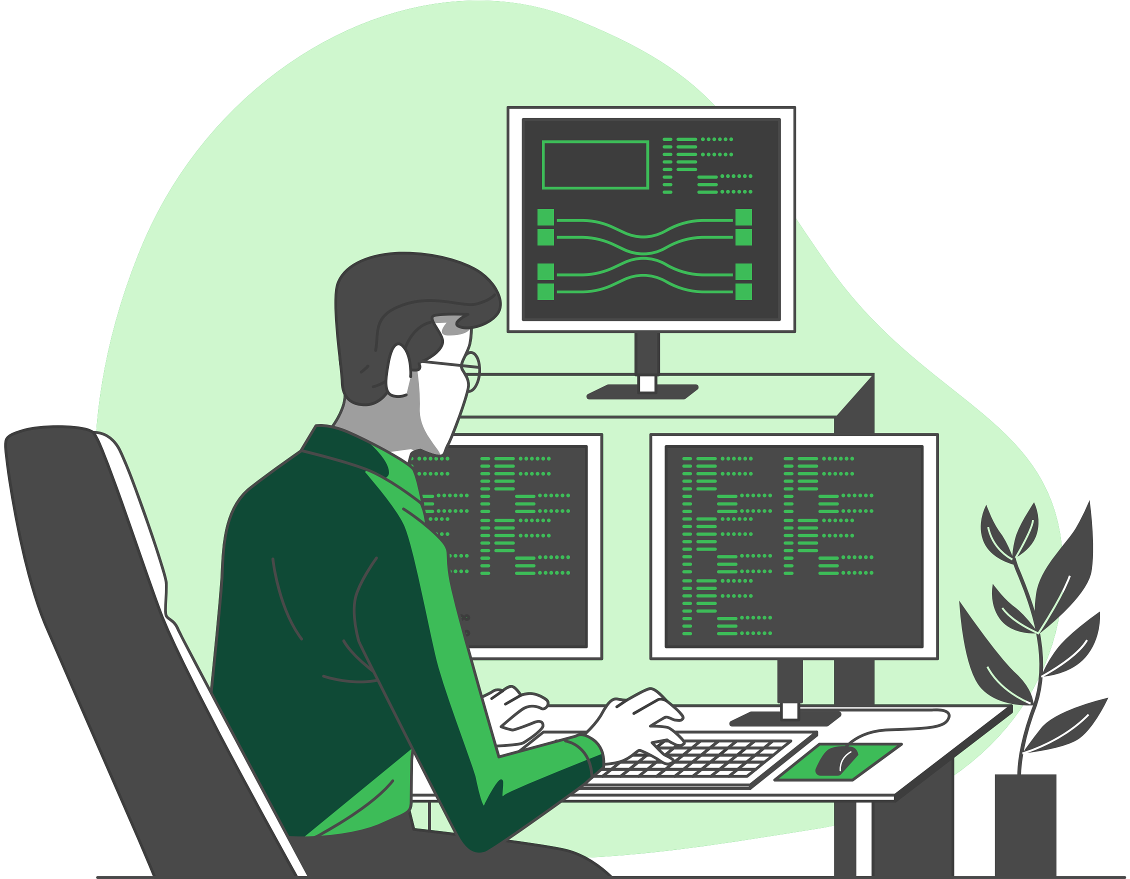
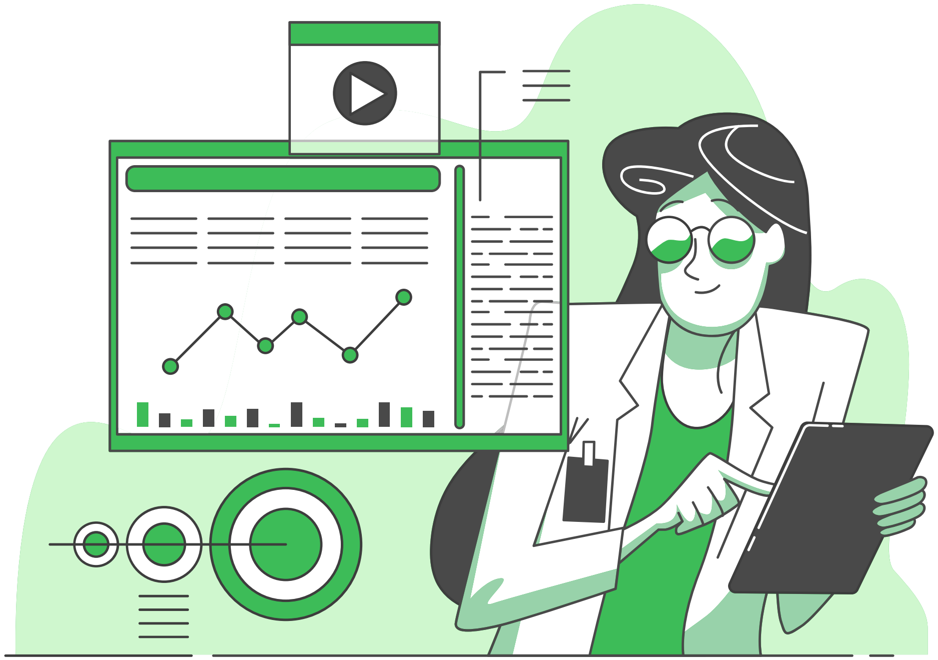


Directed Graph

CS 251 - Data Structures and Algorithms

Note:
Slides complement the discussion in class

Table of Contents



01

Overview

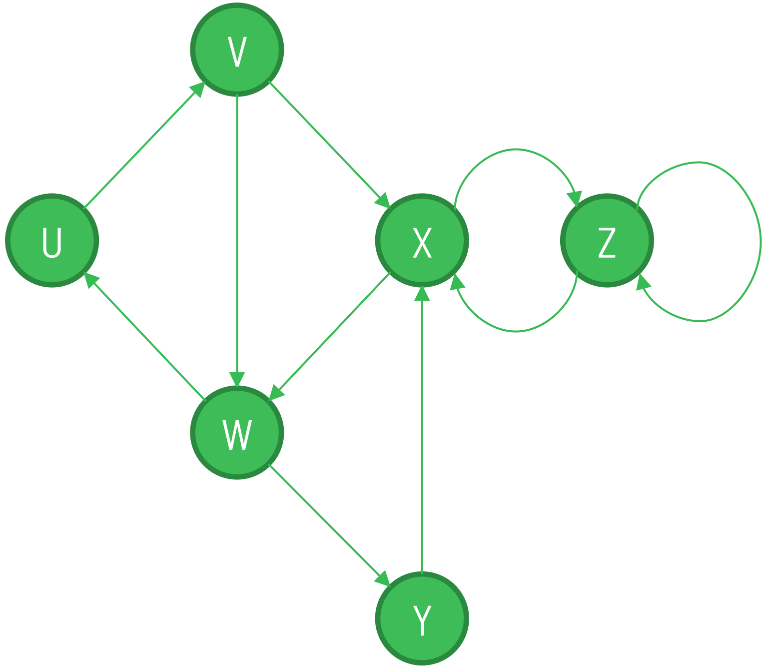
Terminology in the context of directed graphs

Directed Graph

A **directed graph** (AKA. **Digraph**) is a graph whose all edges are directed.

The **in-degree** of a vertex is the number of edges pointing to the vertex.

The **out-degree** of a vertex is the number of edges pointing away from the vertex.



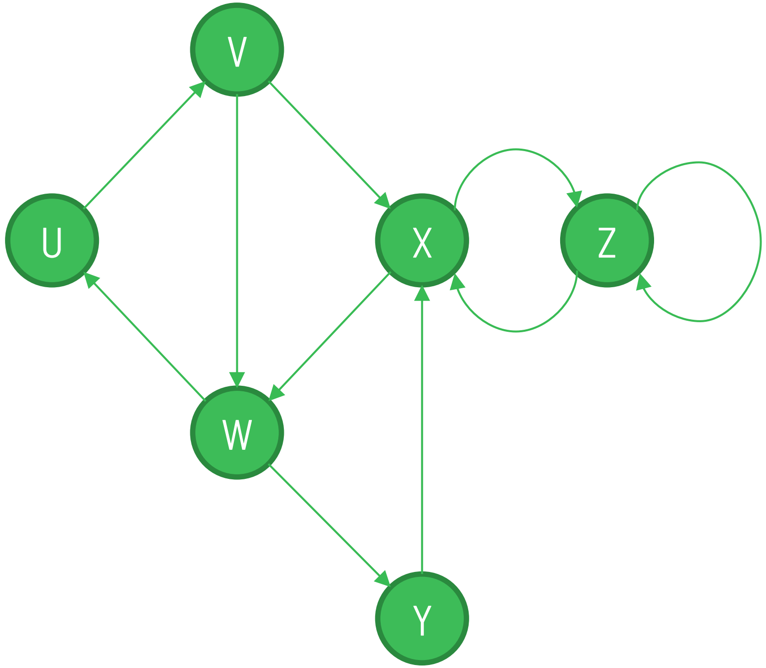
Directed Graph

A **directed path** is a sequence of vertices in a digraph such that there is a directed edge from each vertex to the next vertex in the sequence (e.g., {U, V, W, Y, X})

A **directed cycle** is a directed path whose first and last vertices are the same.

Simple (path or cycle): no repeated vertices or edges.

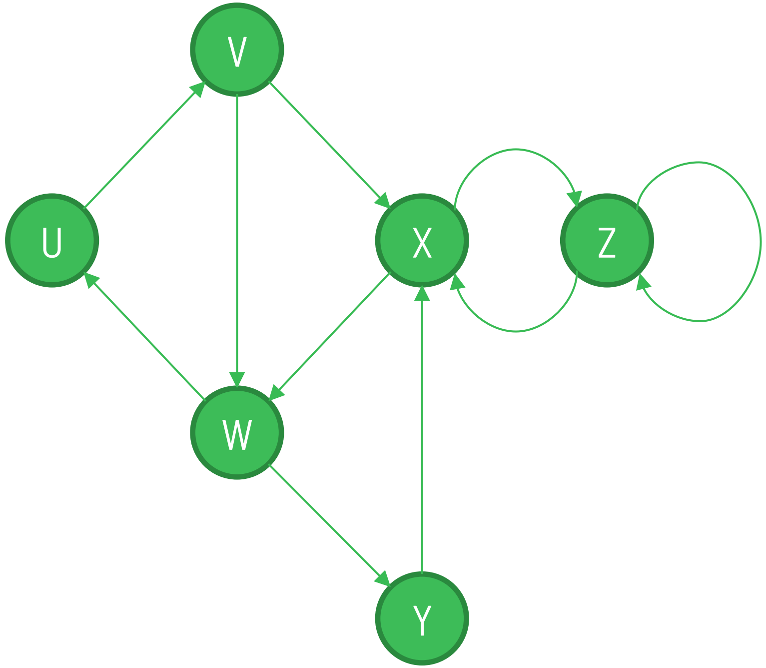
Length (of path or cycle): number of edges.



Directed Graph (Cont.)

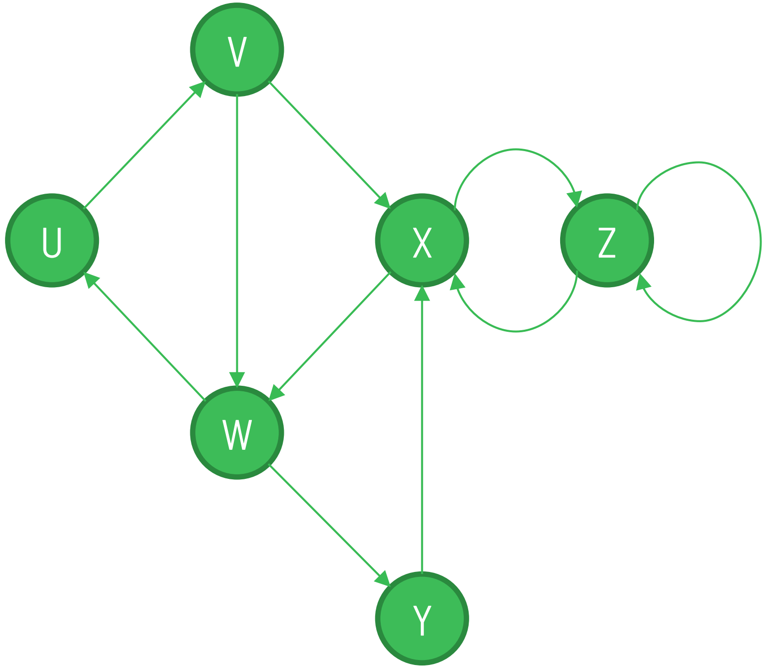
A **simple digraph** has no parallel edges and no self-loops. Note: $(u, v) \neq (v, u)$

In a simple digraph, $|E| \leq |V|(|V| - 1)$



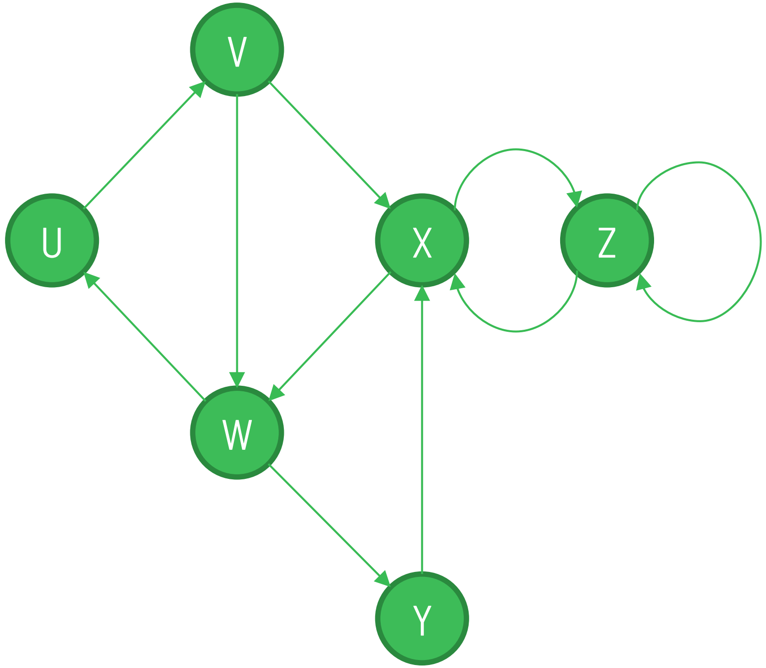
Example: Adjacency Matrix

	U	V	W	X	Y	Z
U	0	1	0	0	0	0
V	0	0	1	1	0	0
W	1	0	0	0	1	0
X	0	0	1	0	0	1
Y	0	0	0	1	0	0
Z	0	0	0	1	0	1



Example: Adjacency List

U	V
V	W, X
W	U, Y
X	W, Z
Y	X
Z	X, Z



02

Reachability

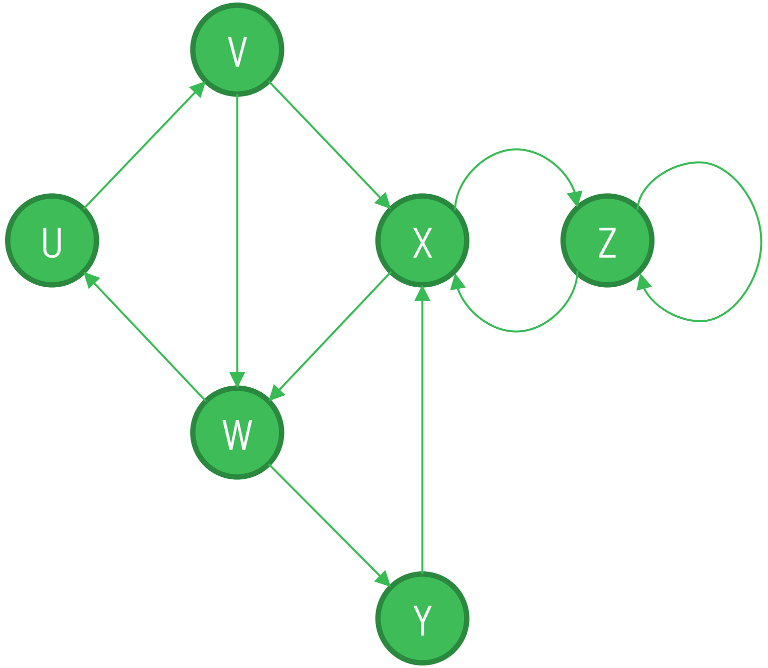
Reaching every vertex

Directed Graph

The edge (u, v) means that the edge is going out of u and into v.

A vertex v is reachable from vertex u iff there is a directed path from u to v.

Example: Is Y reachable from Z?

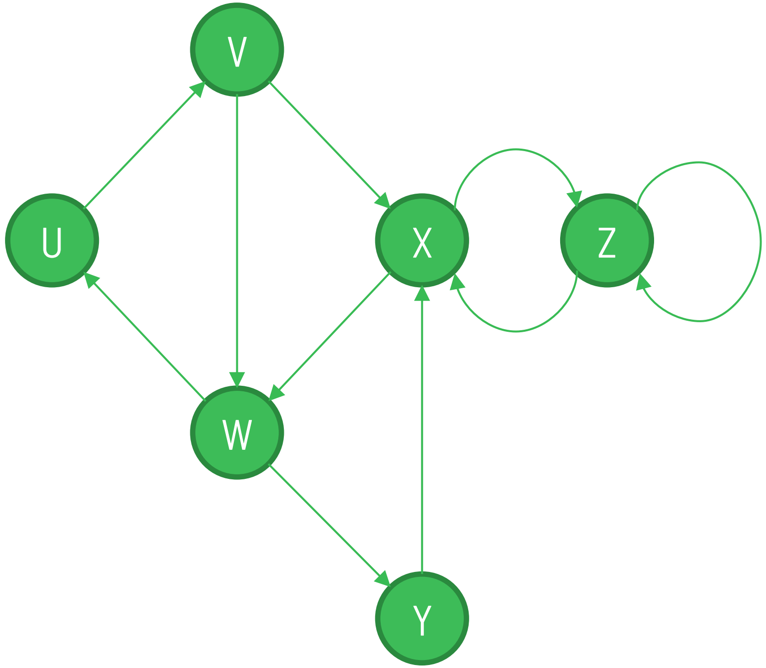


Single Source Reachability

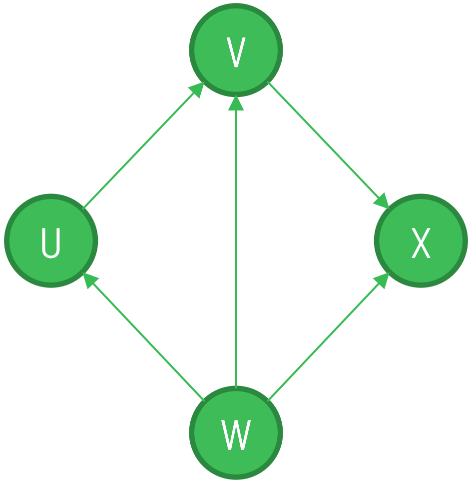
Given a digraph $G = \{V, E\}$ and a source vertex $v \in V$, determine if there is a directed path from v to a destination vertex $t \in V$.

Solution: Use Directed DFS, which is the same old classical DFS but running on a digraph.

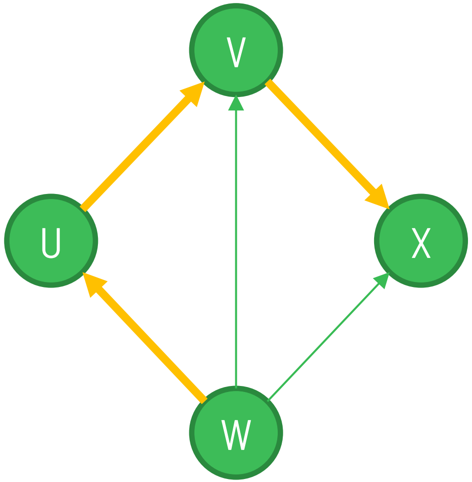
Note: In digraphs, connected does not imply reachability.



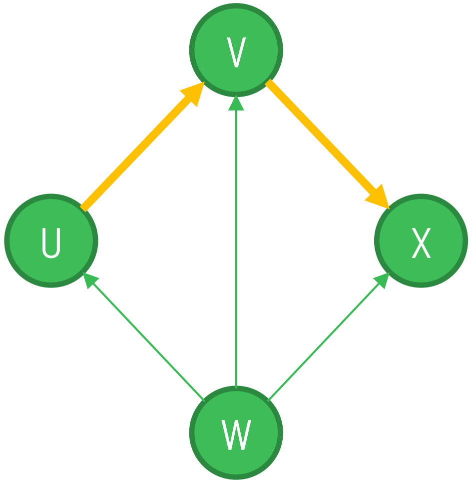
Examples: DFS on a Directed Graph



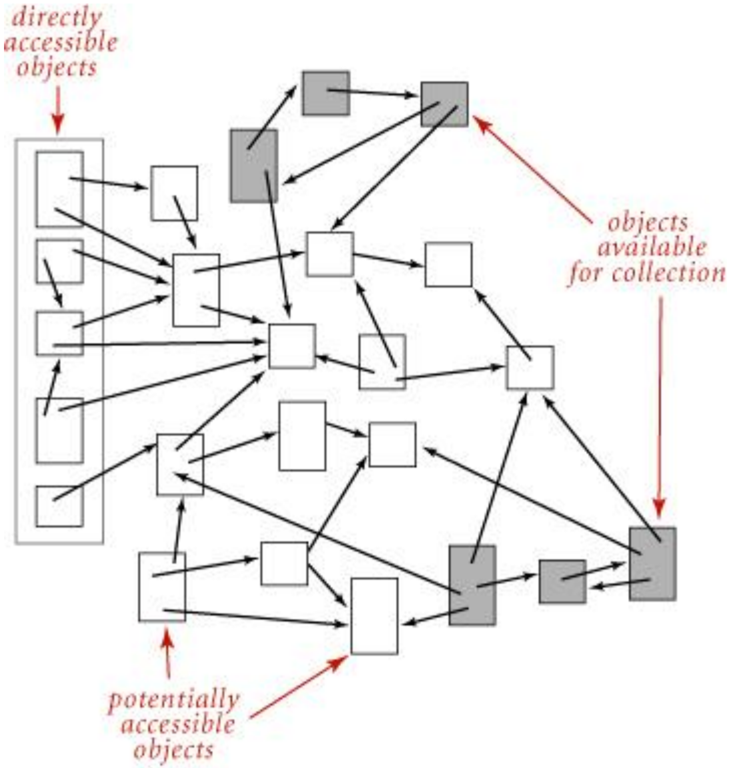
Original graph G



DFS(G, W)



DFS(G, U)



Garbage collection scenario

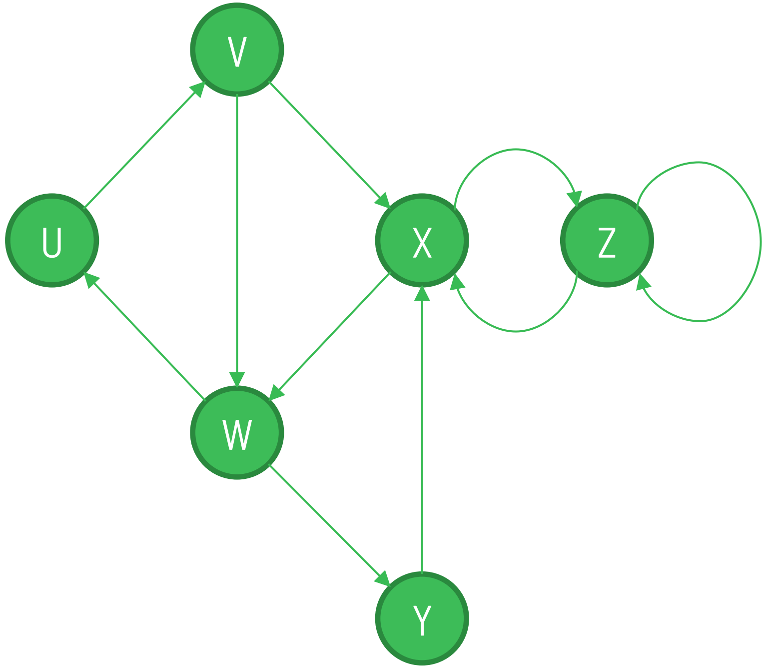
Strong Connectivity

A graph is **strongly connected** if every vertex is reachable from every other vertex (analogous to the idea of connectivity in an undirected graph.)

How can we determine if a graph is strongly connected or not?

Solutions:

- Run DFS on every edge and check all nodes are visited.
- Work smarter, not harder.

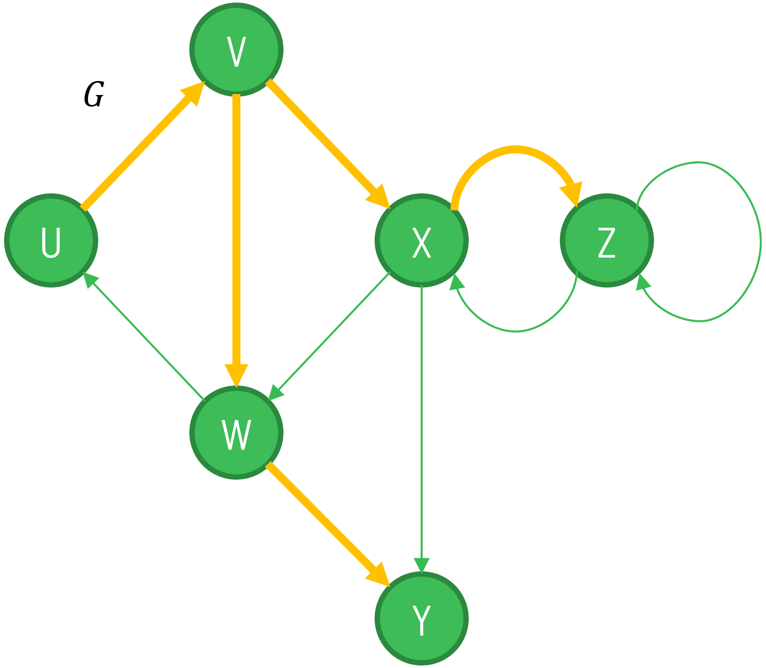


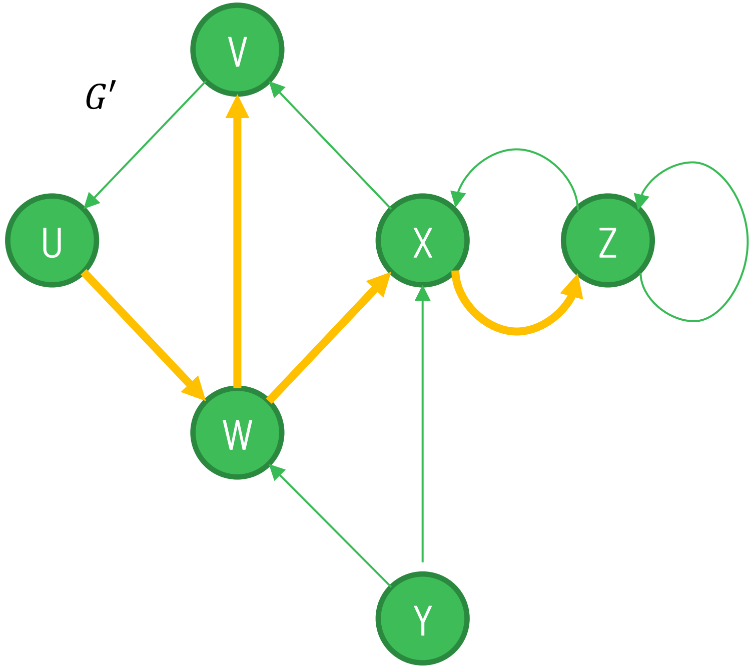
Strong Connectivity Algorithm

- Pick a vertex v in G.
- Perform DFS(G, v)
- If there is a vertex w not visited, then return false
- Let G' be G with edges reversed.
- Perform DFS(G', v)
- If there is a vertex w not visited, then return false. Else, return true.

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

Strong Connectivity Example





Strong Connectivity Properties

Reflexive

Every vertex is strongly connected to itself.

Symmetric

If v is strongly connected to u, then u is strongly connected to v.

Transitive

If v is strongly connected to u, and u is strongly connected to w, then v is strongly connected to w.

We’re Done

Do you have any questions?

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik and illustrations by Stories