

Let's begin at 9:05 PM

L89

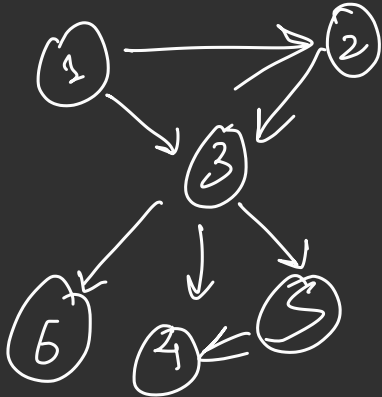
DFS on Directed Graphs & Topological Sorting

Join Discord - <https://bit.ly/ly-discord>

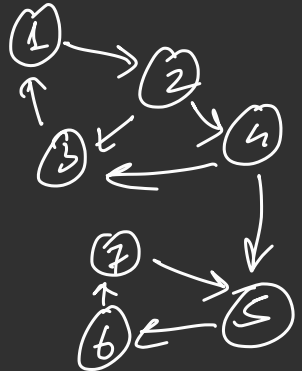
RECAP

## A bit about directed graphs

Directed acyclic \*  
graph (DAG)

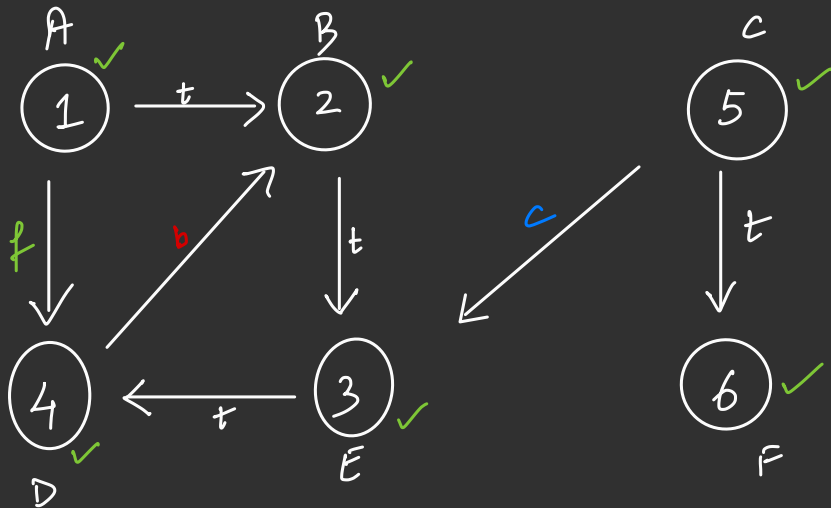


Strongly connected  
components

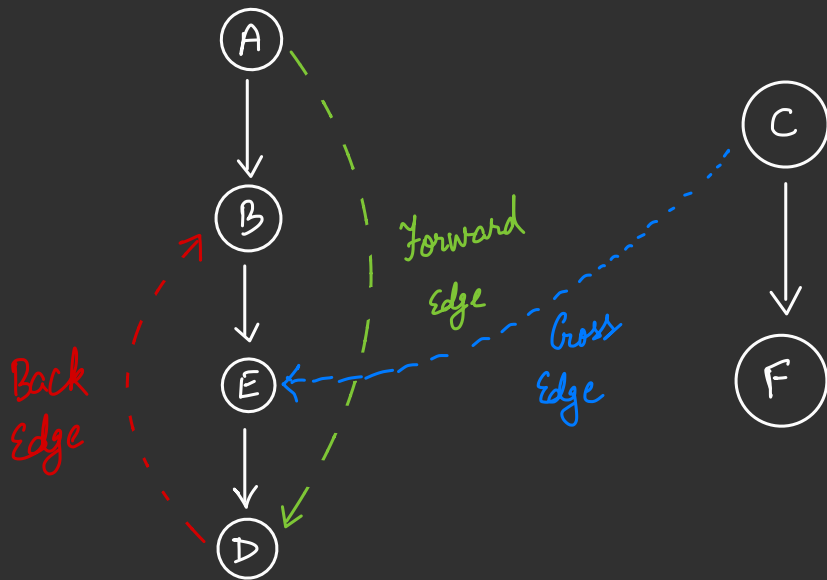


## DFS on Directed Graphs

## Introduction



## Types of Edges



Tree Edge  $\Rightarrow$  Part of DFS tree

Back Edge  $\Rightarrow$  From a descendant to an ancestor

Forward Edge  $\Rightarrow$  Not a tree edge, and is from an ancestor to a descendant

Cross Edge  $\Rightarrow$  From a node in 1 DFS tree to a node in another DFS tree.

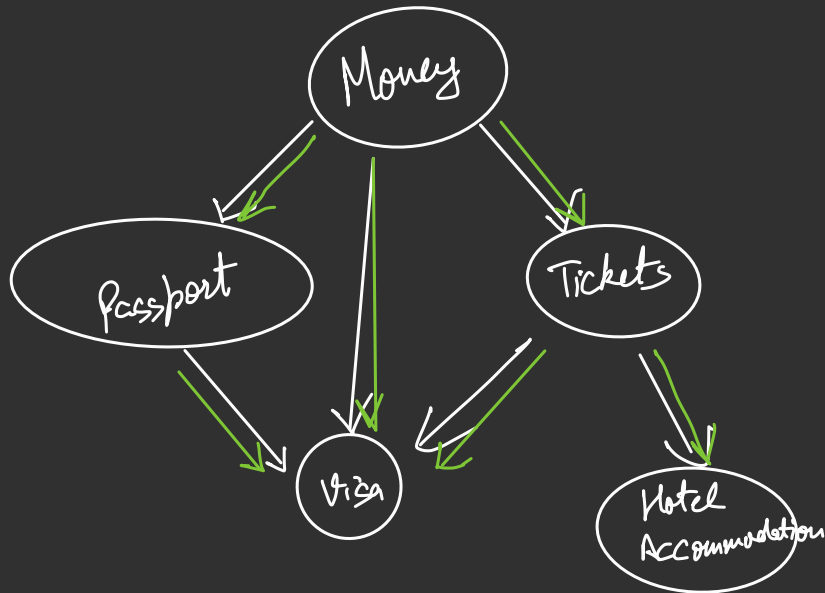
Back Edge  $\Leftrightarrow$

Cycle in  
a directed  
graph



# Topological Sorting

## Intuition (Travel Example)



Money

Passport

Tickets

Visa

Hotel

(Topo Sort)

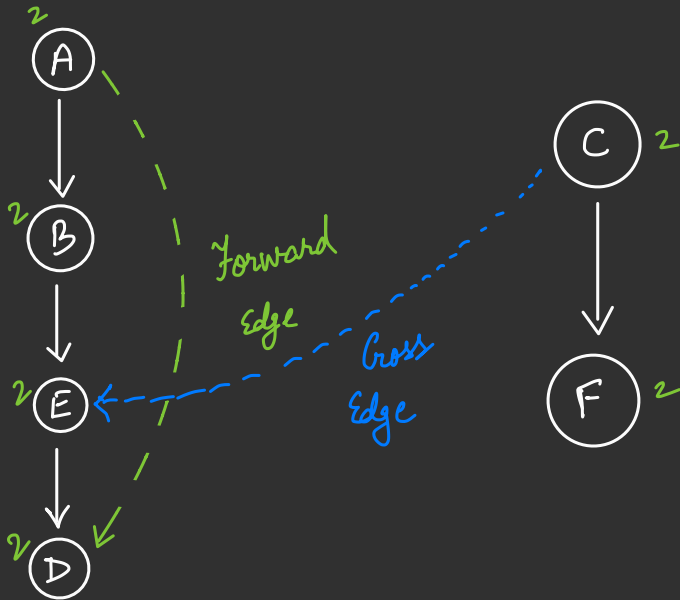
A Topo Sort exists if & only if the given directed graph doesn't have an cycles.

(The graph should be a DAG)

## Introduction

An ordering of the nodes of a DAG  
in such a way that for every edge  
 $u \rightarrow v$ ,  $u$  should come before  $v$   
in the ordering.

How to find?



Reverse Topo Sort

D  
E  
B  
A  
F  
C

Actual Topo Sort

C  
F  
A  
B  
E  
D

Let's Implement

Topological Sort using BFS?

## Practice Problem



## Fox and Names

## Intuition / Solution

$$\begin{array}{rcl} a, b, c & - & \text{---} - - - 2 \\ 0, 1, 2 & - & \text{---} - - - 25 \end{array}$$

$$\text{name}[0] \leq \text{name}[1] \leq \text{name}[2] \dots \text{name}[n-1]$$

$\text{name}[i] = \text{careful}$

$\text{name}[i+1] = \text{car}$

(diff)  $\Rightarrow$  add edge

equal  $\Rightarrow$  <sup>Do</sup> Nothing

$(i+1)^{\text{th}}$  pref of  $i^{\text{th}}$   $\Rightarrow$  Impossible

Let's Implement

# *Thank You!*

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!