CSE221
ALGORITHMS

Topic: Top Sort,
Strongly Connected Components

Prepared by:
Saad Bin Sohan

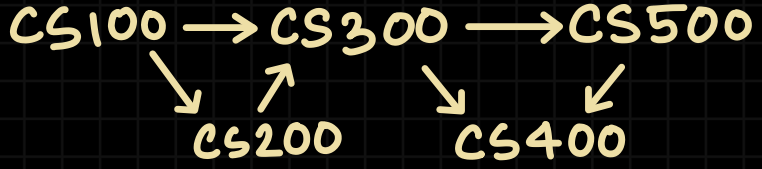
BRAC University

Email: sohan.academics@gmail.com

GitHub: https://github.com/saad-bin-sohan

Topic-1: Tropological Sorting/ Order/Topsort

Consider a student newly admitted to uni. He has to follow a sequence for the courses he can enroll in. Consider a Sequence like this



arrow indicates the order to follow

Now the task is to find all the possible combination of sequences the student can choose from.

Tasks:

→ Graph থেকে order বের করা

Single/Multiple order possible

For example:

এই example এ possible sequences:

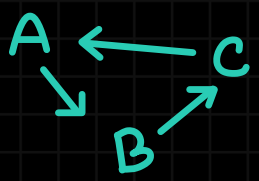
(i) 100, 200, 300
(ii) 200, 100, 300

} reason: both 100 and 200 have no prerequisite to initiate

Que এর graph এর জন্য কি always order বাকা possible?

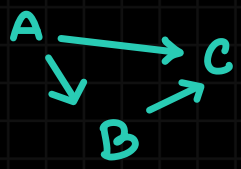
Answer: No.

e.g.



এখানে no specific order can be found because we cannot start any vertex since each of them have pre-requisite resulting in a cycle

but if it was,



we get an order:

A, B, C

since no vertex has to be crossed to reach "A".

Pre-condition to if topsort can be applied:

D (i) Graph টা **D**irected হতে হবে

A (ii) কোনো cycle থাকা যাবে না (**A**cyclic হতে হবে)

G (iii) it has to be a **G**raph

DAG হতে হবে

So if it's a DAG then and only then, topological sorting can be done (initial condition)

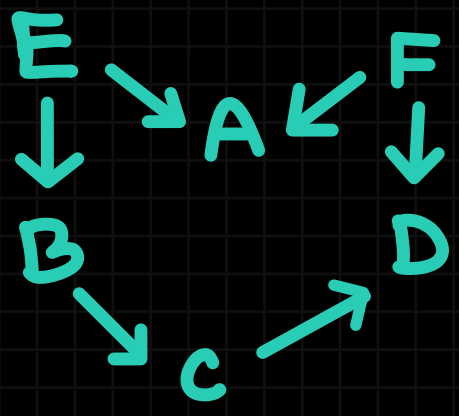
Algorithmic view of Topolgical Sorting

Approach-1: Simple Method

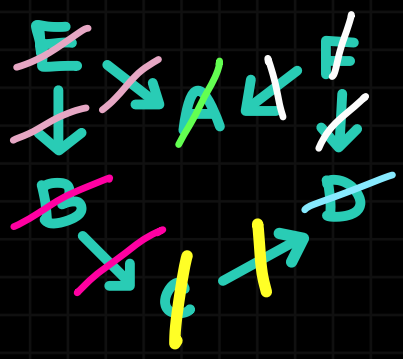
Step-1: Each time Pick a vertex with indegree = 0
(indegree = number of incoming edges)

Step-2: Add that vertex into the sequence; then delete that vertex and its adjacent edges

Que Topsort করো।



solution :



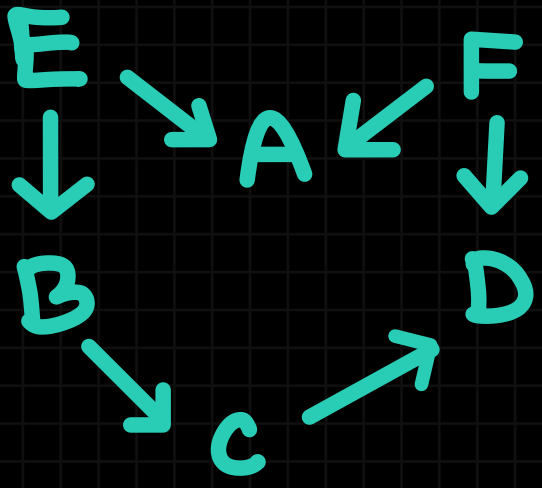
E, F, A, B, C, D

Approach-2: using DFS

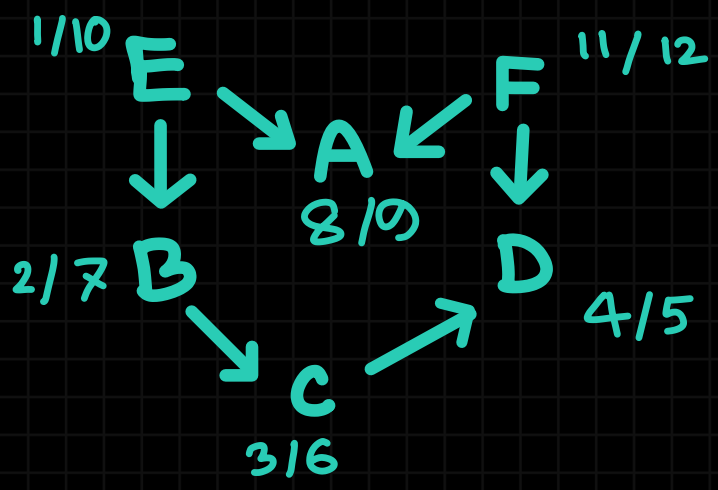
step-1: start DFS at any Vertex

step-2: sort in descending order, according to their finish time

Que Topsort করো (using DFS)



solution:



now we write them in descending order of finish time:

F, E, A, B, C, D

Topic-2: Strongly Connected Components (SCC)

Strongly connected Components is a sub-graph where there

can be a full or partial graph

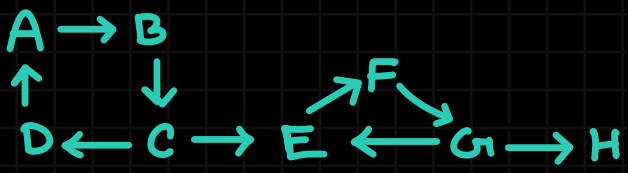
is a path between all pairs of vertices

somehow there has to be a vice-versa way to reach one-another

NB: একটা vertex multiple component এ থাকতে পারে না।

And think of these components as a collection of vertices যাতে প্রত্যেক থেকে বাকি সব vertex এ somehow (not necessarily directly in a single traverse) traverse করা possible

Que



strongly Connected Components কয়টা আছে? 3 টা কী কী?

solution:

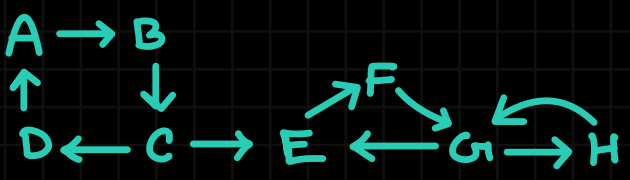
The strongly Connected Components are:

i) A, B, C, D
ii) E, F, G
iii) H

} basically all the elements in a cycle

→ E নাই c ২ E থেকে বাকি সবাই কাছে আসা যায় না।

Que



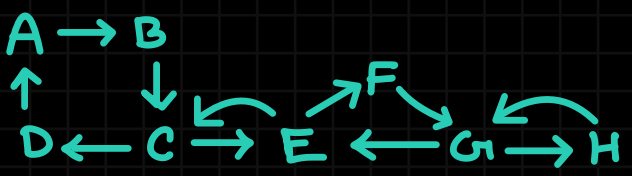
strongly Connected Components কয়টা আছে 3 কী কী?

solution:

The strongly Connected Components are:

i) A, B, C, D
ii) E, F, G, H

Que



strongly Connected Components কয়টা আছে 3 কী কী?

solution:

The strongly Connected Components are:

i) A, B, C, D E, F, G, H

এখানে only একটা strongly connected Component. এজন্য its called "Strongly Connected Graph"

NB: Strongly Connected Graph ≠ Strongly Connected Components

1টা SCC only

Algorithmic approach to find SCC:

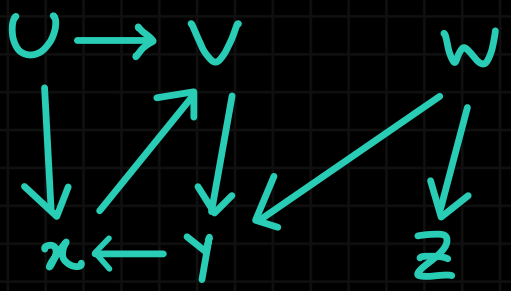
কঙ্কোরাজু Algorithm:

step-1: perform DFS starting at any vertex

step-2: Reverse edges in the given graph

step-3: perform DFS from highest finish time vertex (each time)

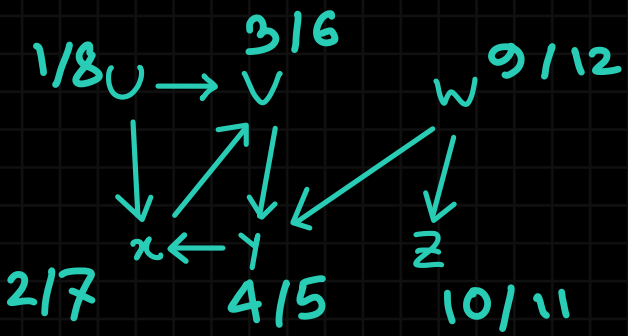
Que



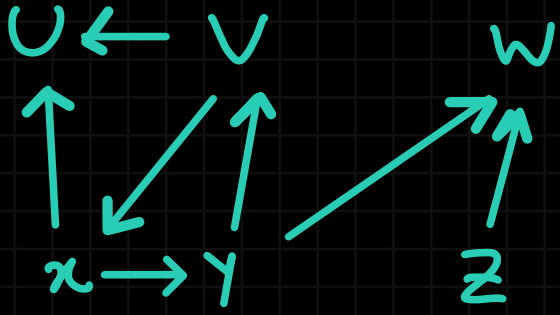
Strongly Connected Components বের করো।

solution:

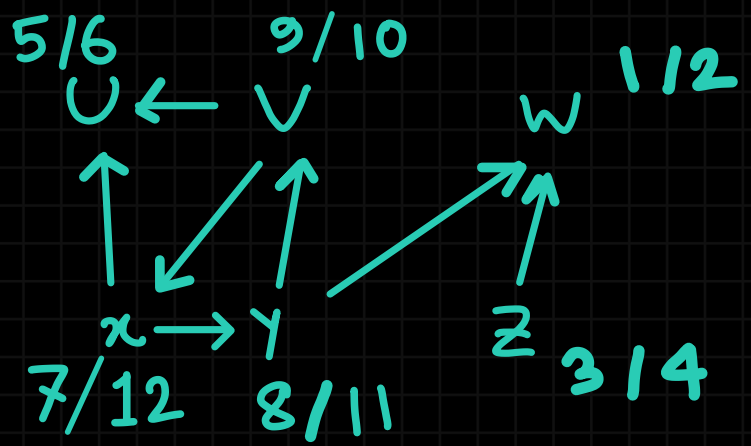
step-1 (DFS on given graph)



step-2 (reversing the edges)



step-3 (DFS on the graph with reversed edges):



The strongly Connected Components are:

i) W

ii) Z

iii) U

iv) X, Y, V

NB: given graph এ we see W, Z, U এদের প্রত্যেকের থেকে কোনো vertex এ যাওয়া এবং somehow ঐ vertex থেকে W বা Z বা U (যেটা থেকে start করেছিলাম) এ return করা possible না। That's why they are alone as strongly connected components.

But X, Y, V create a cycle and so make up a single strongly connected component.

Final এ question type:

Scenario তে vertex রা নিজেদের মধ্যে কাজ করে বা particular কোনো কিছু share করবে নিজেদের মধ্যে। There we see how that has strongly connected components.