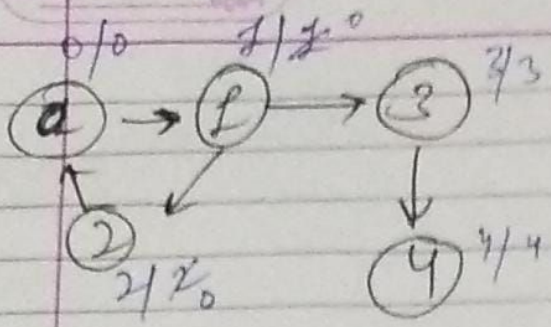
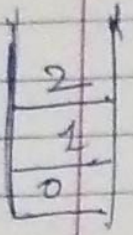
① Strongly connected components



(Tarjan's Algo)

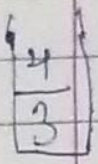
(I)



when (0/0) Point comes
Pop from stack until
0 found.

[2, 1, 0] → SCC 1

(III)



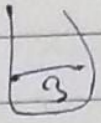
when (4/4) comes

Pop until 4 comes

→ do white Type

[4]

(IV)



when (3/3) Comes

Pop until 3 comes

[3]

② Longest Path in DAG
$
Shortest Path (Topological sort)
(Problem solving)

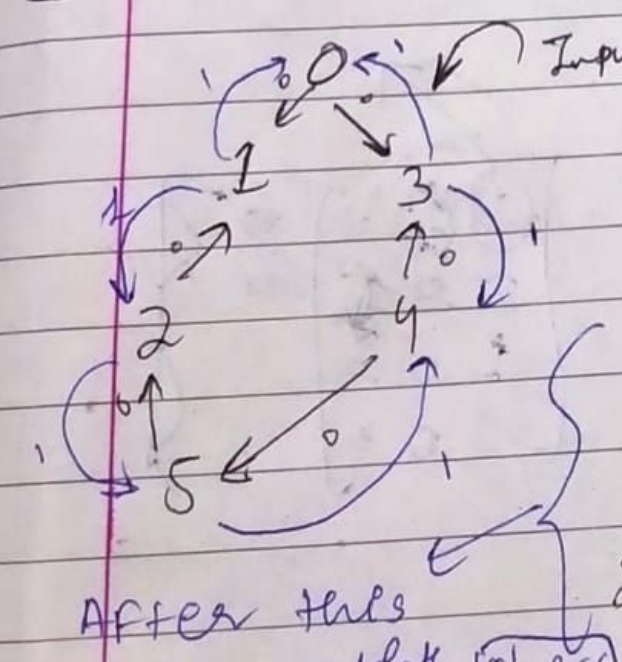
1) Get Topological order
2) Loop through the topological order (u) — v_1, v_2, v_3 } neighbours
$ see the minimum distance

if (dis[u] != INF &&
dis[u] + wt < dis[v])

---

② Making wired connections Problem
③ Minimum edges to Reverse to make Path from source to destination
① Graph coloring (M-coloring)

Minimum edges to Reverse to Make a Path from source to Dest:



1) Provide a '0' weight to original edges
2) Create Reverse edges with weight 1

(3) Number of operation to make network connected

n = 4 , [[0,1],[0-2],[1-2]]

1) Disjoin union set concept is used.

→ DSU dsu = new DSU(n);

So ab your connect the comput ers Reduce them $ don't waste the cable.

0 — 1 ← P(0) != P(1)

1

2

↙

P(0) != P(2)

Comp --
⇒ 4-1 = 3
⇒ 3-1 ⇒ 2

For connecting 3 comp Require 2 edges (cables)

ans ⇒ Comp-1

Graph Coloring : Chromatic no. min color req to color a graph.

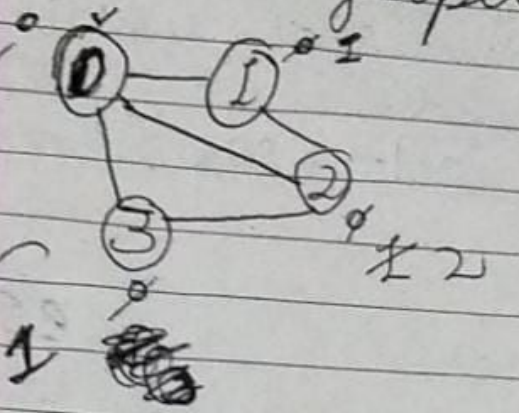
M-coloring Problem :

can we color a graph using M colors.

Approach → BFS (not to optimal)

Q1: Color a graph → BFS
Q2: M coloring → Backtracking

(I) color a graph (BFS)



1) Take node from queue & check all adjacent color if same increase there color
2) If adj nodes not visit add them to queue.

(II) M-coloring Prob. (Backtracking)