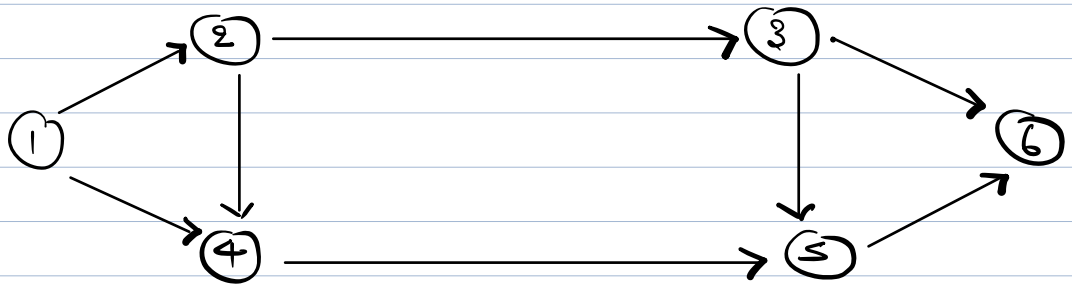
Given N dependent jobs. find the order in which these jobs can be performed.



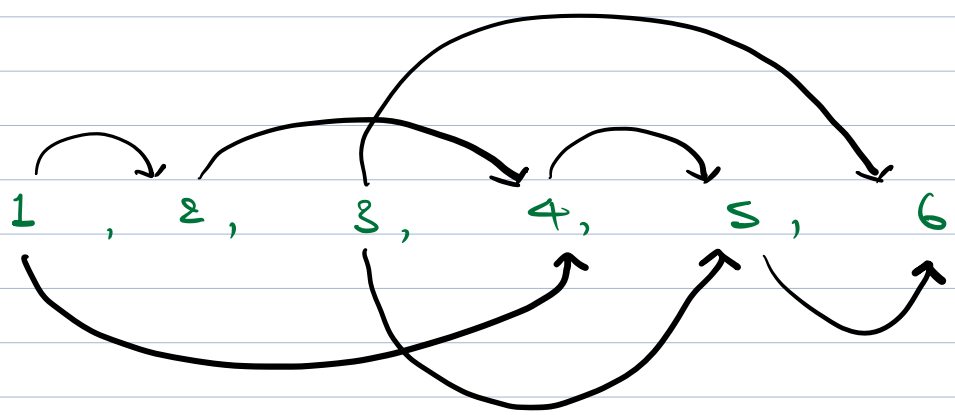
(1, 2, 3, 4, 5, 6) or (1, 2, 4, 3, 5, 6)

A → B (B is dependent on A)

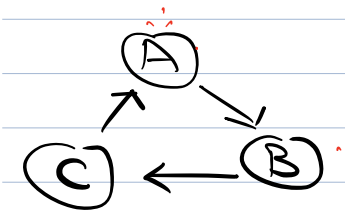
perform A before you can perform B.

Topological Sort

⇒ Linear ordering of nodes. s.t. if there is a path from the node i to the node j then i is on the left of j



Directed Acyclic Graph



(Deadlock)

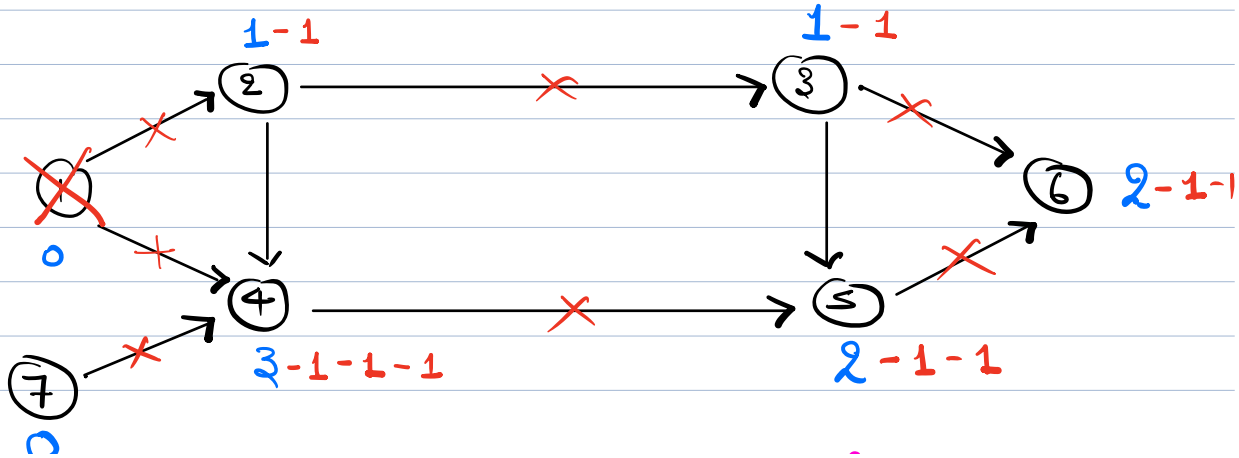
P1 → P2

4 Ways to Perform Topological Sort

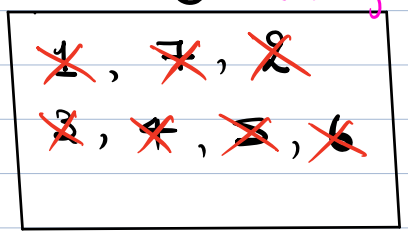
1) Using Indegree of A Node

↓

No. of incoming edges.



Set (Degree 0)



1 0
2 1 0
3 1 0
4 3 2 1 0
5 2 1 0
6 2 1 0
7 0

1, 2, 3, 7, 4, 5, 6

Code

↓ Degree [N+1];

1 → 0
2 → 0 1
3 → 0 1
4 → 0 1 2 3
5 → 0 1 2
6 → 0 1 2
7 → 0

→ 1 → 2, 4
→ 2 → 4, 3
→ 3 → 5, 6
→ 4 → 5
→ 5 → 6
→ 6 → x
→ 7 → 4

```
list < int >   order;
Queue < int >  degreeZero;

for (i = 1; i <= N; i++) {

        if (Degree[i] == 0) {

                degreeZero . enque (i);

        }

}

while (! degreeZero. isEmpty()) {
```

```
int x = degreeZero.dequeue();
order.add(x);
for (all u connected to x) {
    Degree[u]--;
    if (Degree[u] == 0) {
        degreeZero.enque(u);
    }
}
```

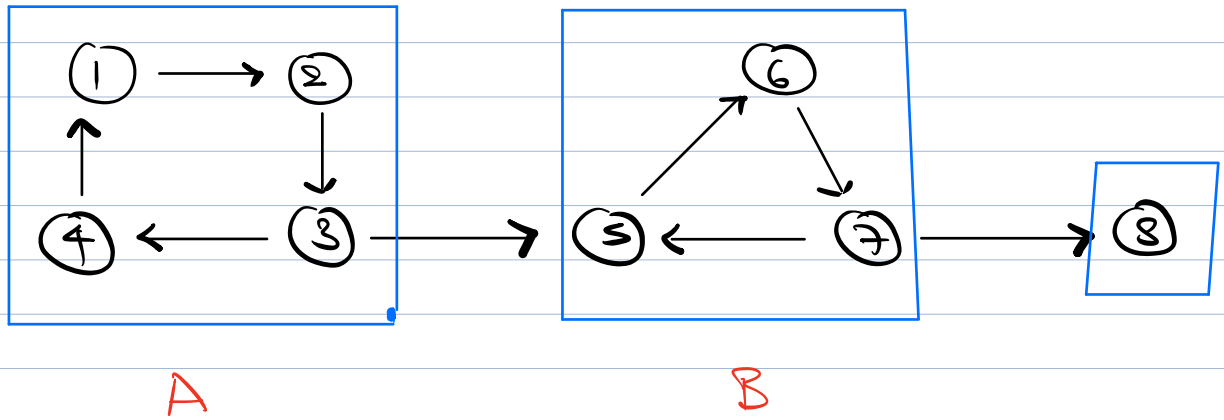
$$T.C. = O(V+E) + O(V+E)$$

$$= O(V+E)$$

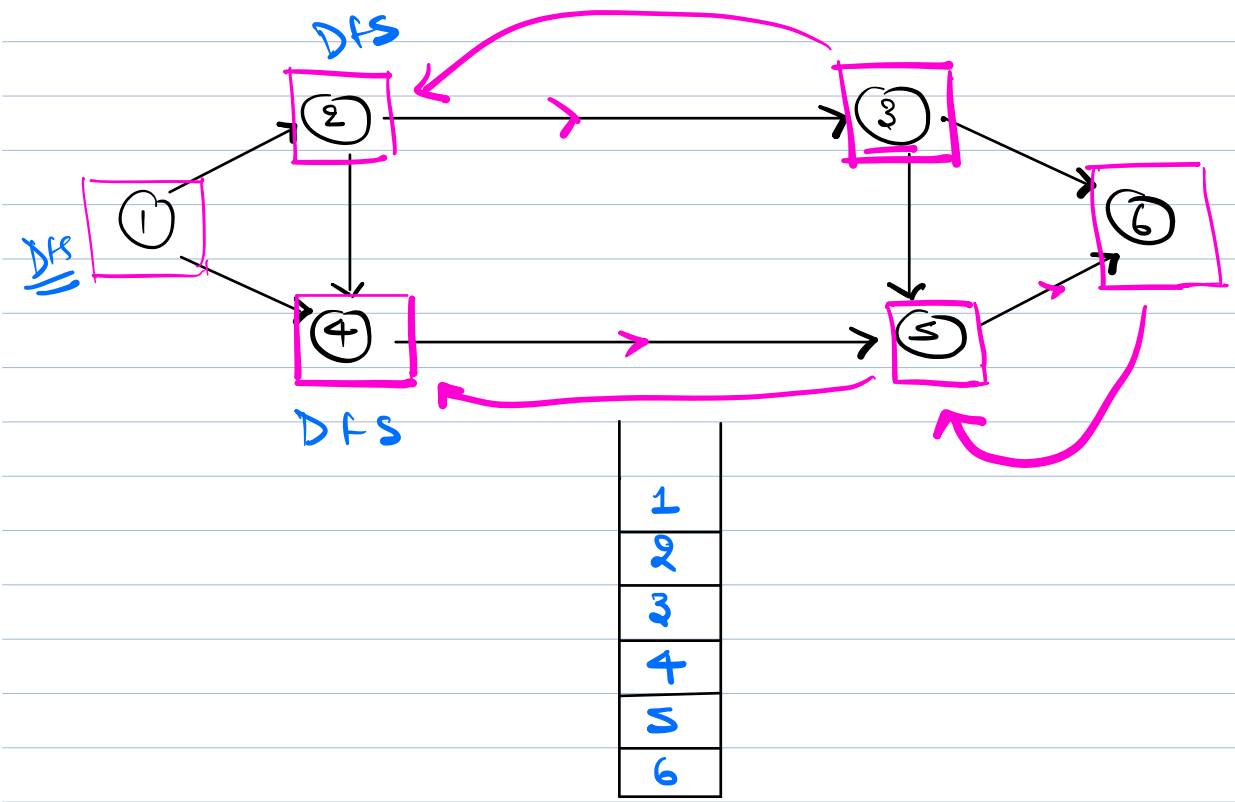
Strongly Connected Component

Path-graph

⇒ Path from every node to every other node.

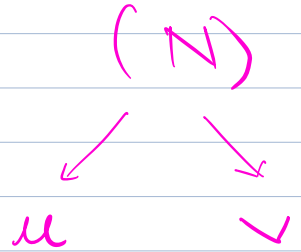


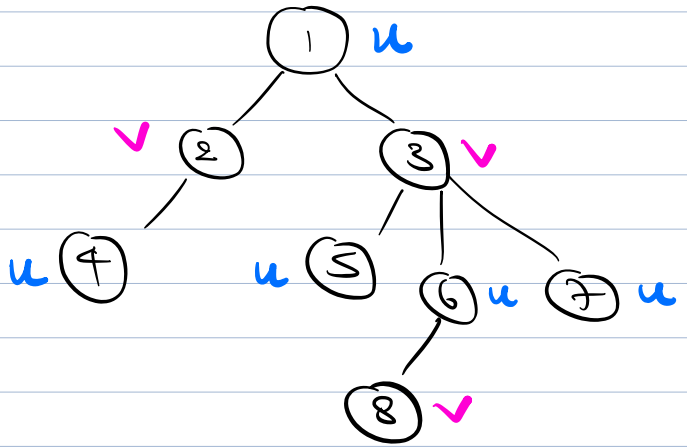
2) Using Stacks (Outdegree)



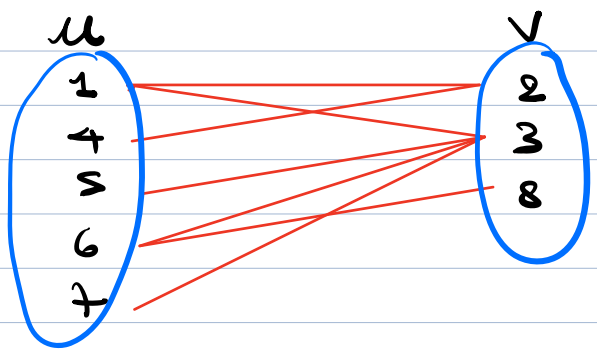
T.C. = $O(V+E)$

✡ Bipartite Graph





⇒ Bipartite graph

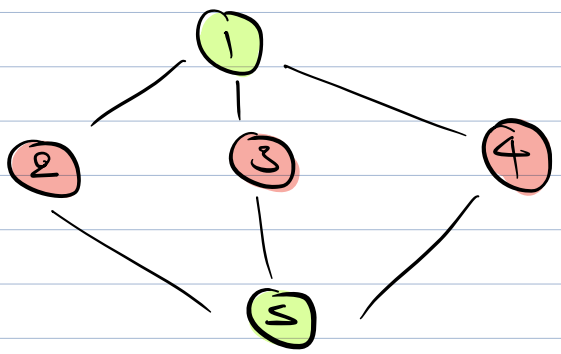


⇒ A graph whose vertices can be divided into 2 disjoint & independent sets (u & v) s.t. every edge connects a vertex in u to a vertex in v.

Given a graph, Identify if it is bipartite ??

✱ Graph Coloring

⇒ Color the nodes of the graph s.t. no. to adjacent nodes have the same color. Using min no. of colors.

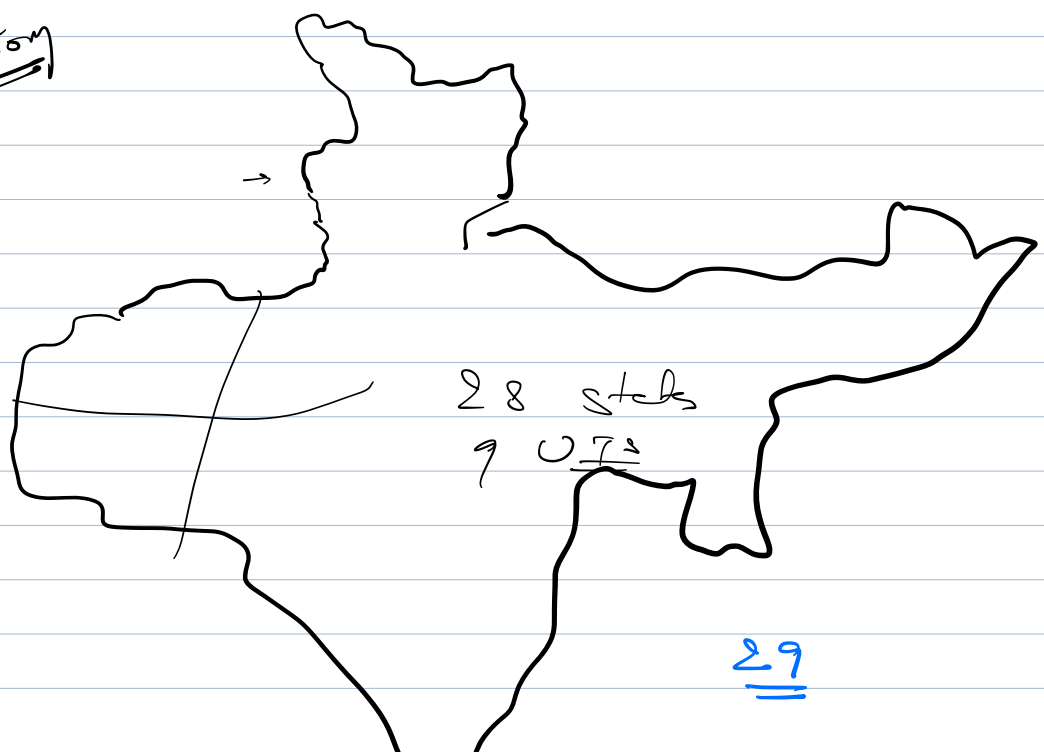


(2)

⇓

Bipartite

History



29

1852 → European Researches

⇒ 4 colors to color a map.

Augustus De Morgan

↳ Set Theory

↳ Mathematical induction.

✓ ✗ ✓ ✗ ✓ ✗

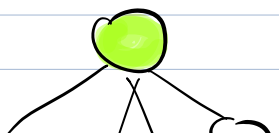
124 years. ⇒ first proof supported by a computer using a simulator.

1990 →

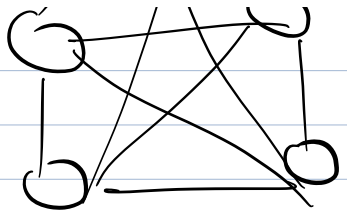
Chromatic No.

↳ Min no. of colors req. to color a graph.

Fully Connected N nodes

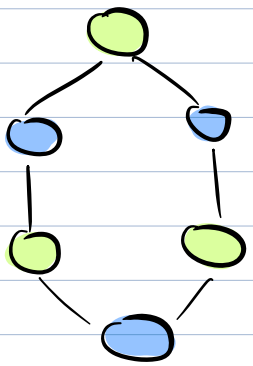


(N)

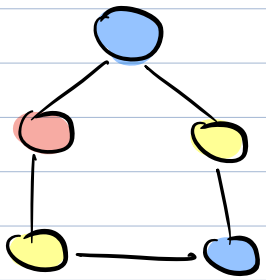


(Back Tracking)

Chromatic no. of a cycle ??



(2)



(3)

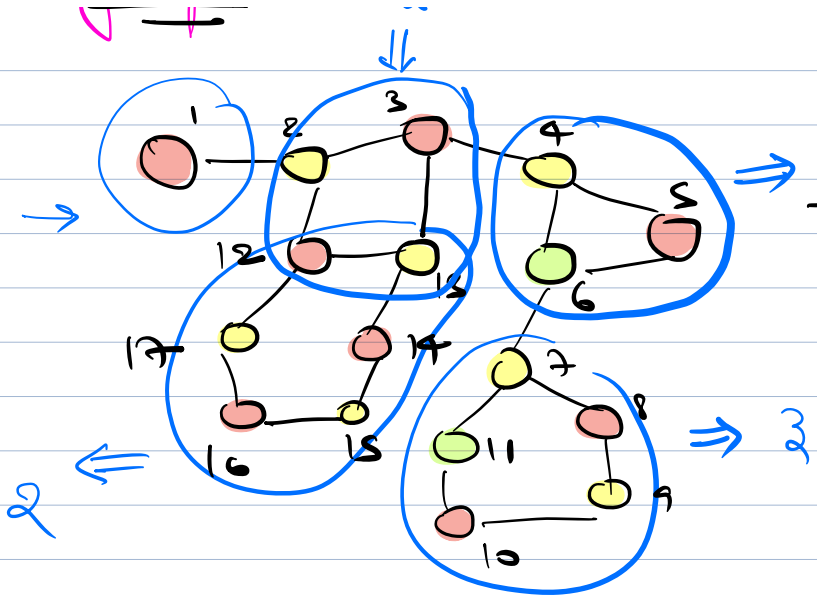
nodes (Even) = 2

nodes (Odd) = 3

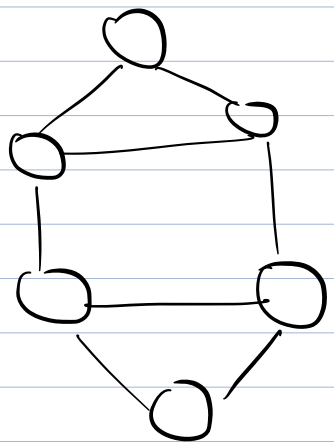
Chromatic no. of a tree ??

= 2

Graph 2



No Odd length Cycle ⇒ Bipartite graph

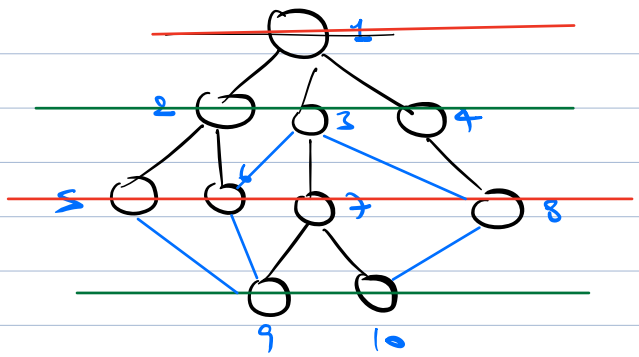


H.W. Calculate the length of every cycle.

Hint: Maintain all nodes in the path (DFS)

Q Given a tree with N nodes.

Find the max no. of edges that can be added to this tree so that it remains bipartite.



U (odd)

1
5
6
7
8

X

V (even)

2
3
4
9
10

Y

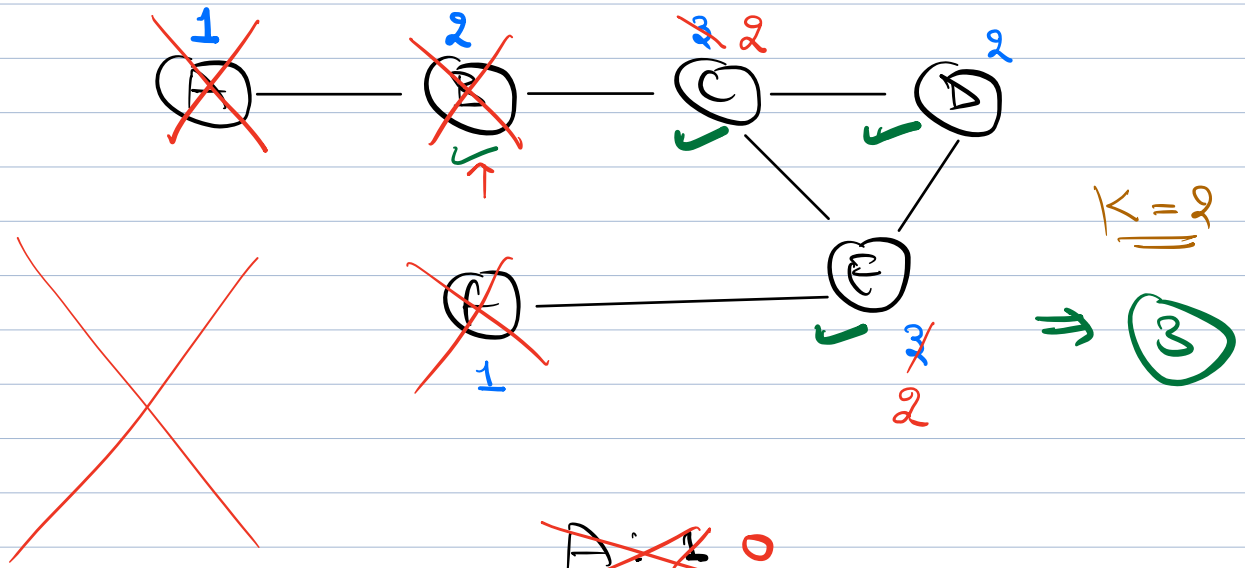
$(X \times Y) - (N-1)$

N people who want to attend a party.

You can only attend a party if K of your friends are already the party.

Max no. of people that can attend the party

Given friendship graph (Undirected)



A : ~~1~~ 0

B : ~~2~~ 1 < K

C : ~~3~~ 2

D : 2

E : ~~3~~ 2

F : ~~1~~

~~A, K, B~~

PS Session (Last Session)