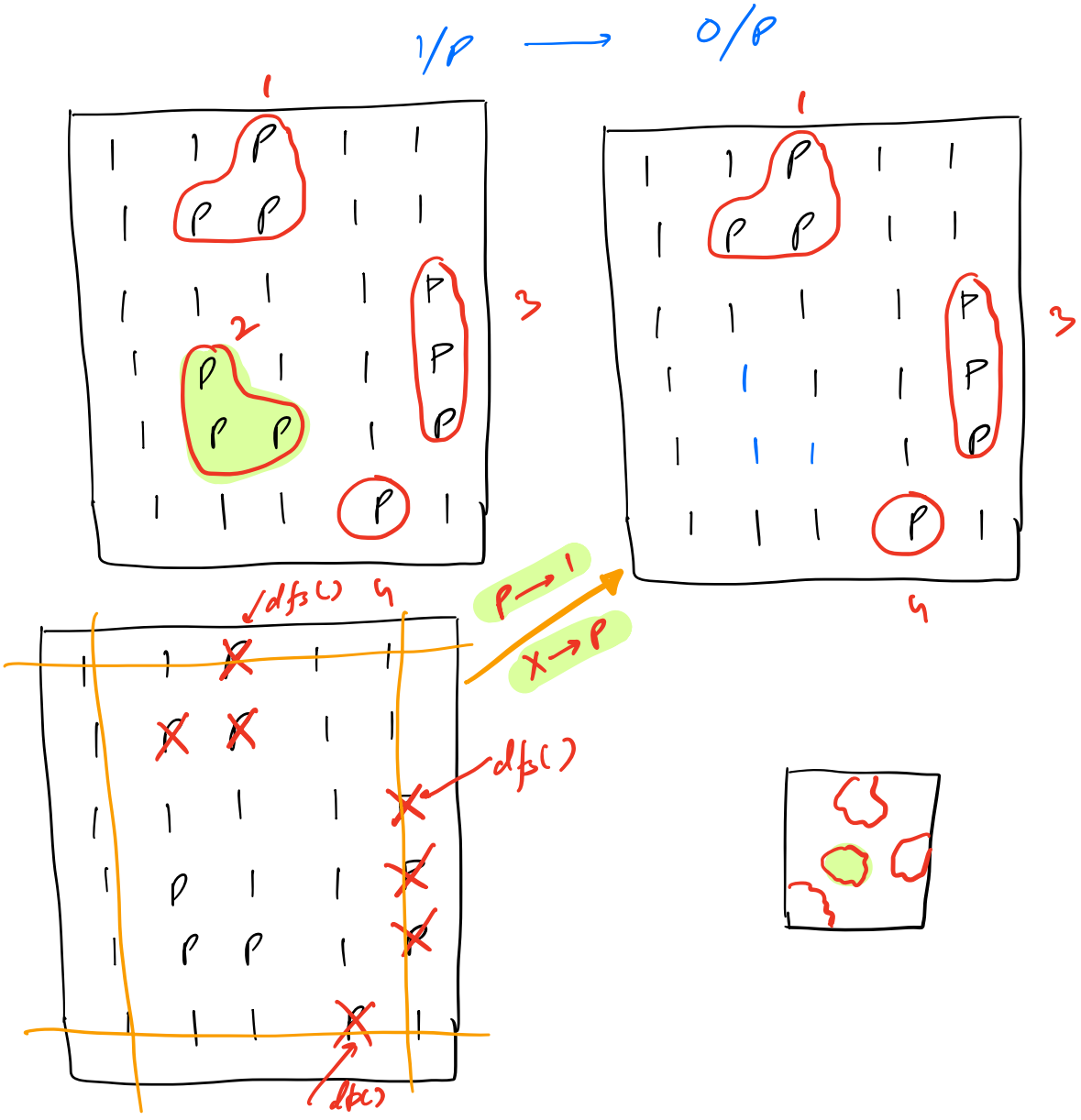
Q. Given the state of a map after war b/w I & P.
Initially it was all I.
P attacked & occupied some region.

I is a peaceful country, so it will take back only that land which is completely surrounded by I.
Find the state of the map after I takes back the land.



dfs()

$TC = O(V+E) + O(NM)$

$V \rightarrow NM$

$E \rightarrow NM$

$TC = O(NM)$

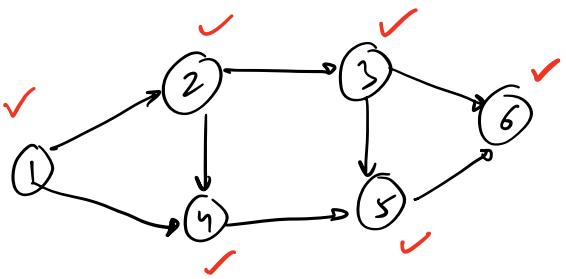
$SC = O(V)$

$SC = O(NM)$

1) ~~visited[][]~~ : if a cell is 'X' ⇒ it is visited!

2) stack

Q Given N dependent jobs.
Find out the order in which you can perform these jobs.

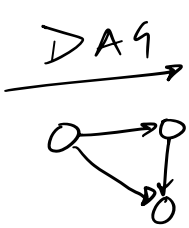


A is dep. on B

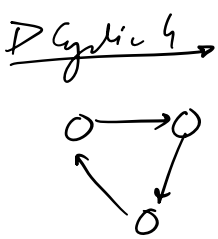
B, A

→ 6 5 3 4 2 1

→ 6 5 4 3 2 1



✓



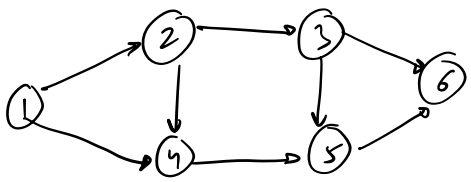
X

⑧ Topological Sorting

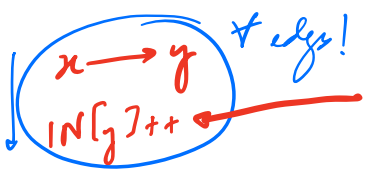
→ Linear ordering of nodes s.t. if there is a path from node i to node j, then i should come before j in the order!

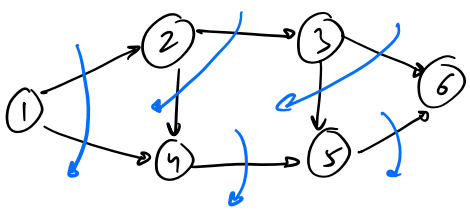
→ 1 2 3 4 5 6

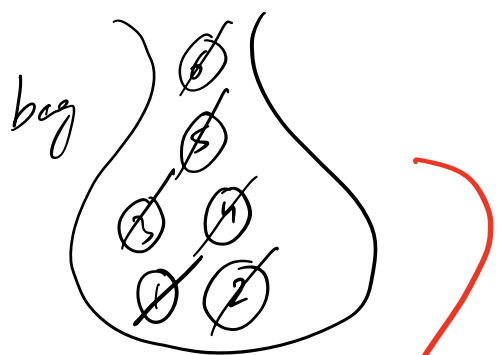
→ 1 2 4 3 5 6




```
int IN[N+1] = {0};
```







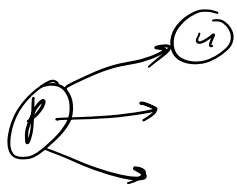
IN

1	0
2	X 0
3	X 0
4	X X 0
5	X X 0
6	X X 0

TOPO: 1 2 4 3 5 6 →

```
bag [] ;
// Create the IN[]
f(i = 1 ⟶ N)
    if( IN[i] == 0)
        bag.add(i);

while( ! bag.isEmpty()) {
    int p = bag.back();
    bag.pop_back();
    ANS.push_back(p);
    f( u : adj[p]) {
        IN[u]--;
        if(IN[u] == 0){
            bag.add(u);
        }
    }
}
ret ANS;
```

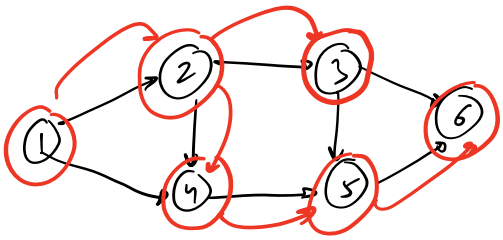


⟶ if ANS.size() != N
⟹ Cycle !

TC = O(E) + O(V) + O(V+E)

SC = O(V)

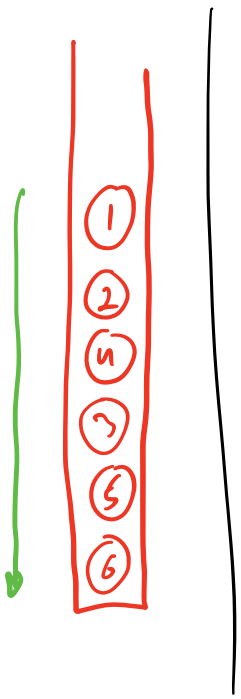
TC = O(V+E)



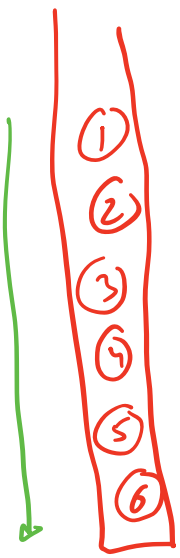
dfs(3)

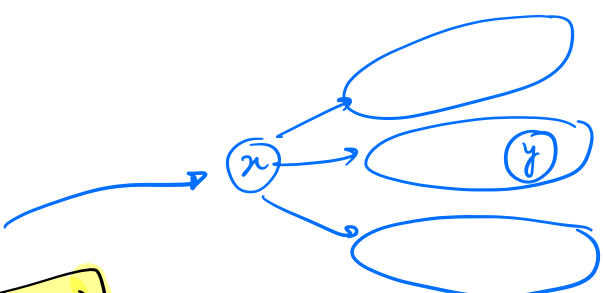
dfs(2)

dfs(1)



dfs(1)





TC = O(V+E)

SC = O(V)

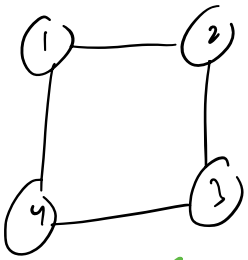

```
f(i = 1 ⟶ N)
    if(!vis[i])
        dfs(i)
```

TOP ⟶ reverse(ANS)

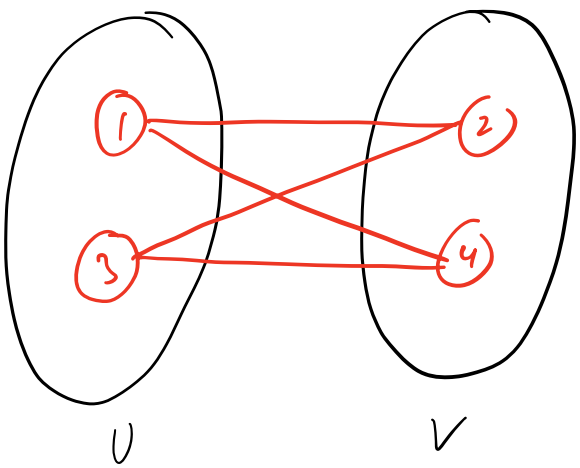
```
dfs(v){
    [ ]
    ANS.push_back(v);
}
```

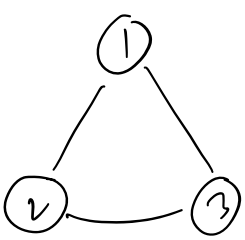
⑧ Bipartite graph

A graph which can be broken down into 2 disjoint sets U & V s.t. no 2 nodes in U or no 2 nodes in V have an edge b/w them!

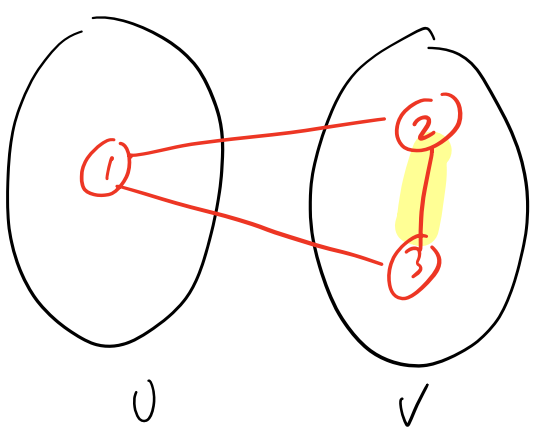


✓ bipartite

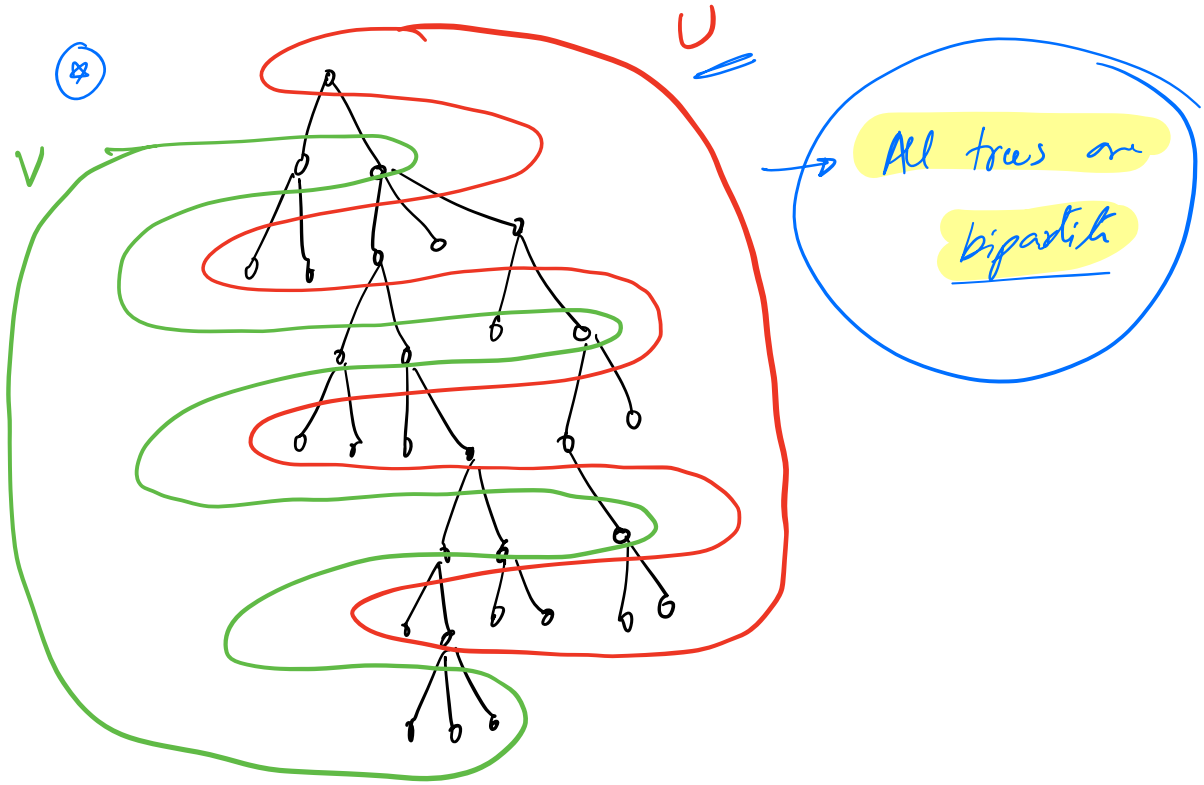


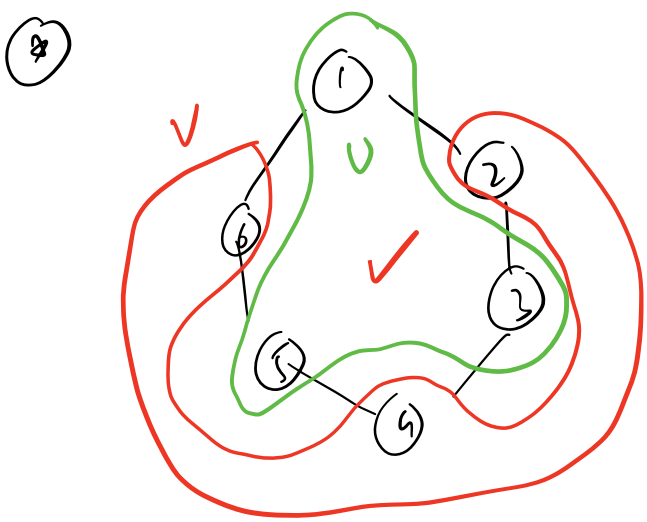


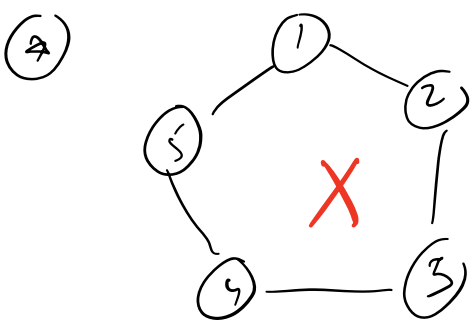
X Not bipartite



Q Given a graph, check if it is bipartite!







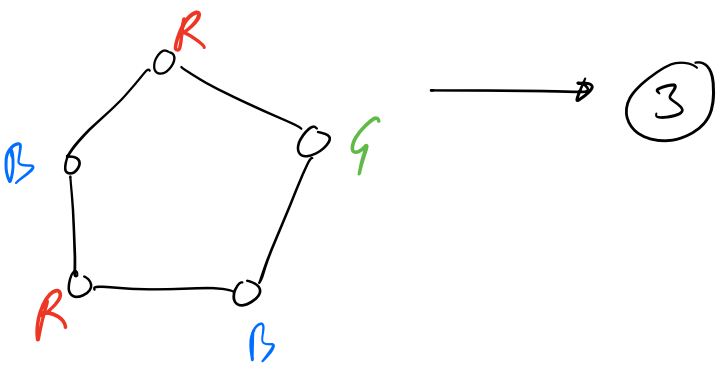
All even len cycle are bipartite

All odd ——— not ———

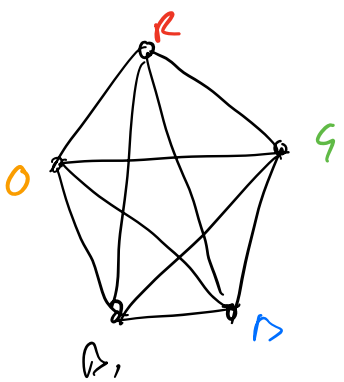
Graph Coloring →

④ Chromatic no:

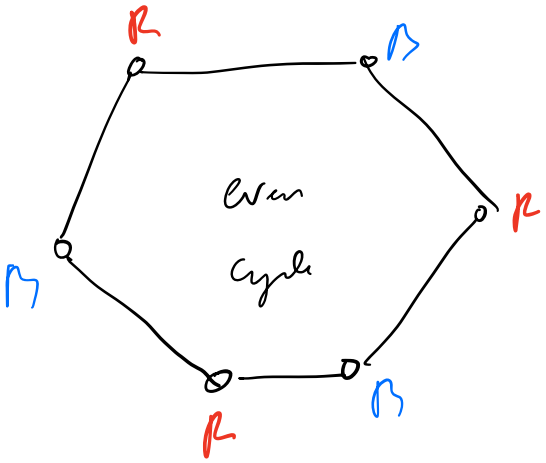
Min no. of colors req'd to color a graph s.t. no 2 adjacent nodes have the same color!



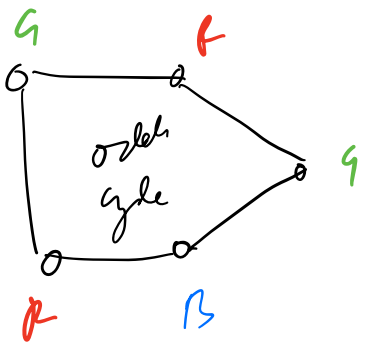
Fully Connected graph →



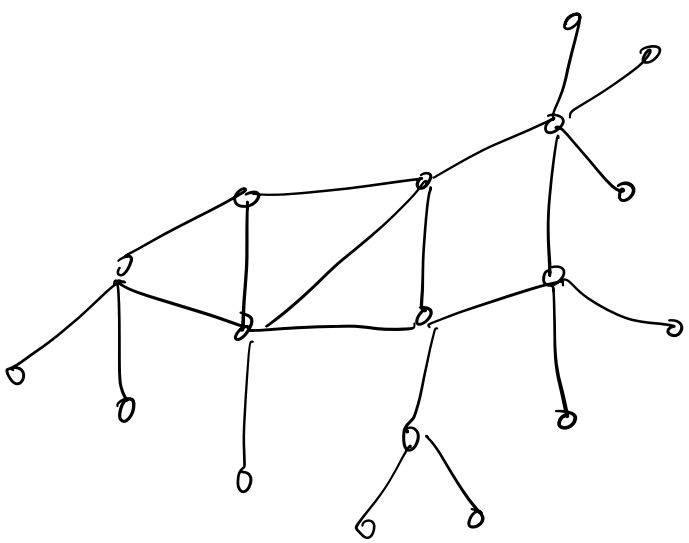
Ch. no : V



ch No : 2



ch. No : 3



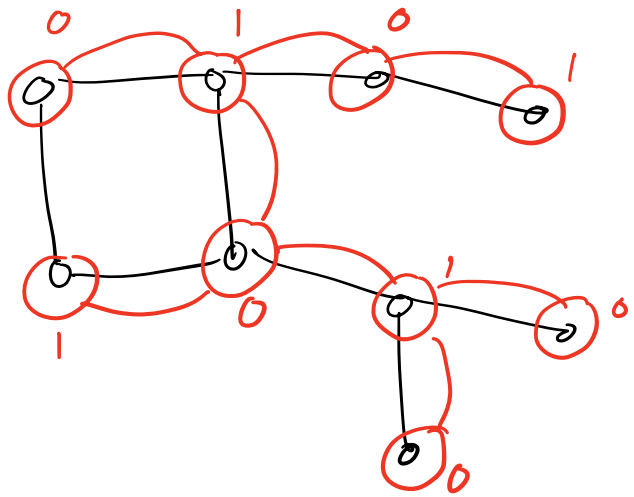
ODD cycle $\Longleftrightarrow$ ch No > 2 $\Longleftrightarrow$ NOT bipartite

0/1

Ch. NO <= 2

NO ODD cycle

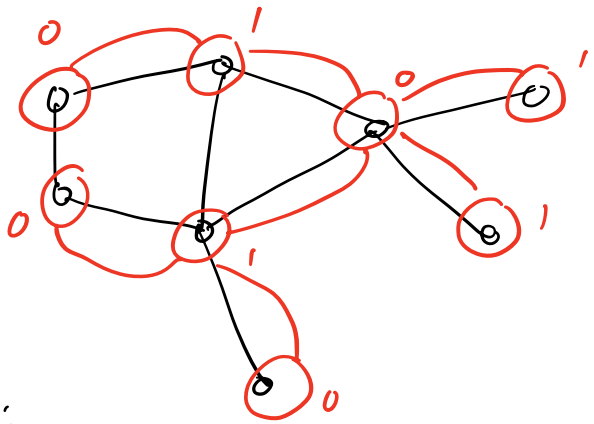
BIPARTITE Graph



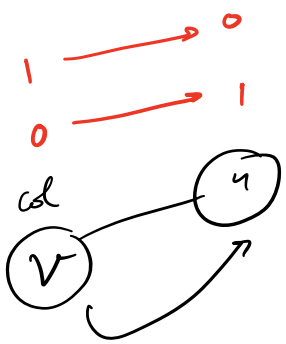
Ch. NO > 2

ODD cycle

NOT bipartite



```
int c[N] = {-1};
bool vis[N] = {false};
bool ok = true;
void dfs(v, col) {
    vis[v] = true;
    c[v] = col;
    for(u : adj[v]) {
        if(vis[u] == false) {
            dfs(u, ____);
        }
```



```
clr {
    if ( c[u] == col ){
        ok = false;
    }
}
}
}
```

TC = O(V+E)

SC = O(V)

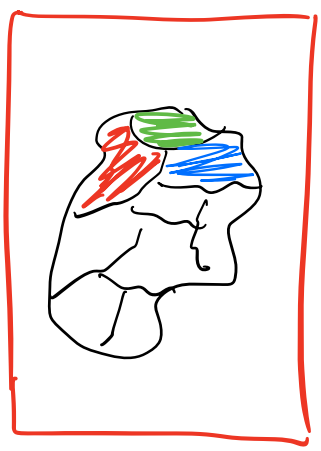
MAP COLORING

ATLAS

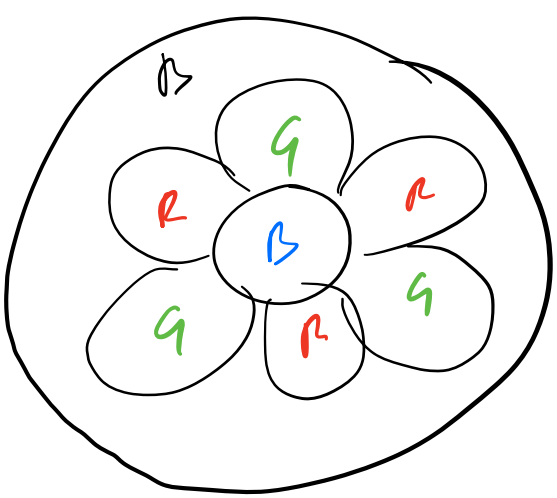
~1800

1852

≤ 4 colors ✓



brother: Augustus – de-morgan



X