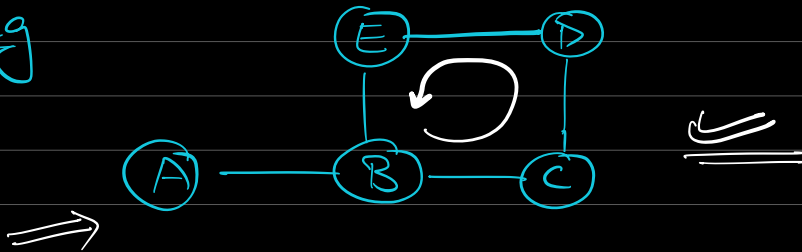


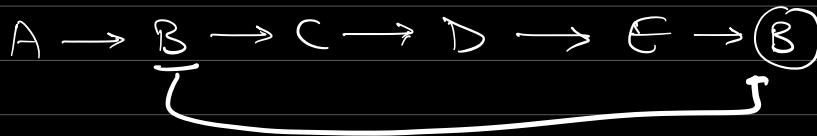
Q Given an undirected graph, Check if there is a cycle or not.

Eg



Solⁿ

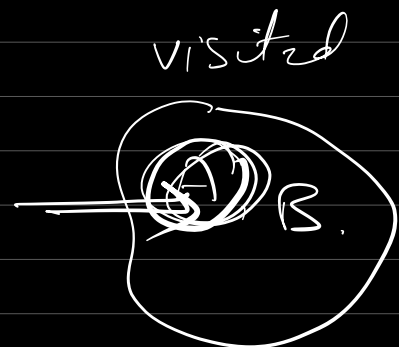
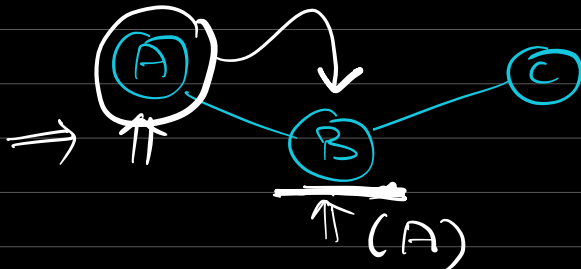
DFS



Basic Intuition

⇒ Start traversal from any given node & mark visited

⇒ If you reach a visited node again, there is a cycle.

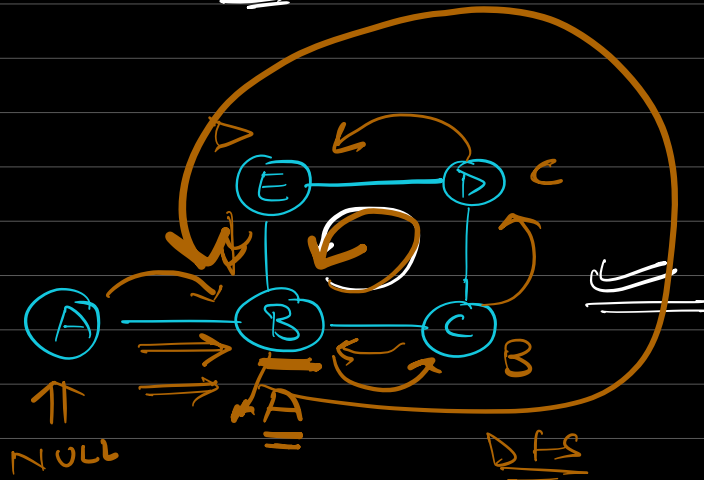


parent

$B \Rightarrow \underline{\underline{A}}, \underline{\underline{C}}$

$\underline{\underline{D}}, \underline{\underline{E}}$

DFS



A, B
C, D
E

$B \Rightarrow \underline{\underline{A}}, \underline{\underline{C}}, \underline{\underline{E}}$

$E \Rightarrow \underline{\underline{B}}, \underline{\underline{D}} \times$

$C \Rightarrow \underline{\underline{B}}, \underline{\underline{D}}$

$D \Rightarrow \underline{\underline{E}}, \underline{\underline{C}}$

Code

bool isCyclic(v, parent) {

\Rightarrow visited[v] = true;

Neighbor \Leftarrow for (all u connected to v) {

AM \Rightarrow iterate over row m[v] if (visited[u] == true && u != parent) {
return true;

AL \Rightarrow list[v] else if (!visited[u]) {

bool cycle = isCyclic(u, v);

```
if (cycle == true) {  
    return true;  
}
```

Y

3

return false;

5

 \vee, \in

T.C. $\Rightarrow O(\underline{V+E})$

Why DFS \neq not BFS??

Q

2 countries

I

5

P

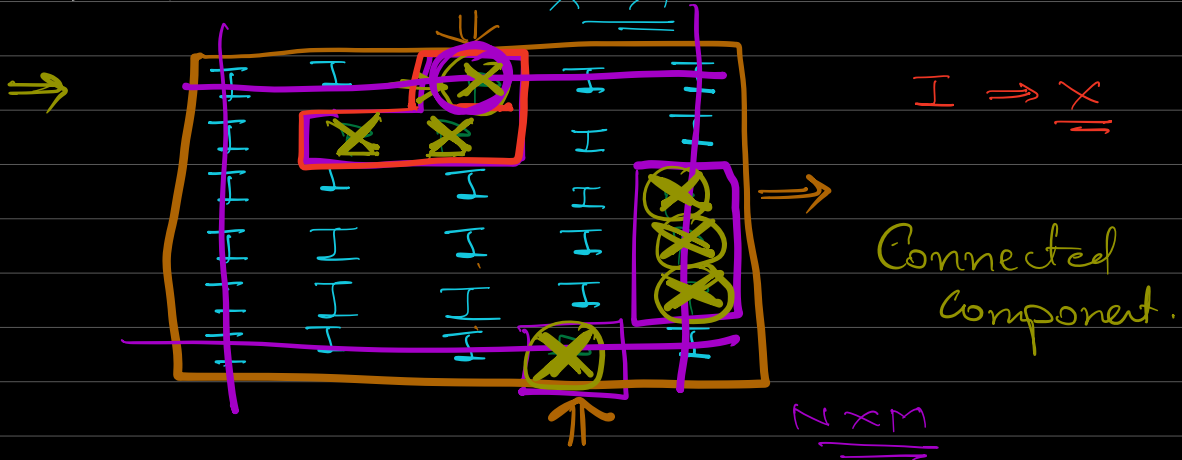
二

11

Peaceful
country

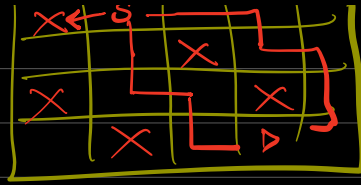
motorious

2D Matrix

$$\underline{X \Rightarrow P}$$


P loser





$$T.C. \Rightarrow O(3NM) \Rightarrow O(NM)$$

\Rightarrow Since identifying P's which we will convert back to I's is difficult

\Rightarrow We mark all the P's which we will not convert.

\Rightarrow All these regions will have atleast one node present on the boundary.

\Rightarrow Iterate on the boundary of the matrix

& from each P, start a BFS/DFS

to mark the P's of that connected component.

\Rightarrow Convert remaining P's to I's.

Google

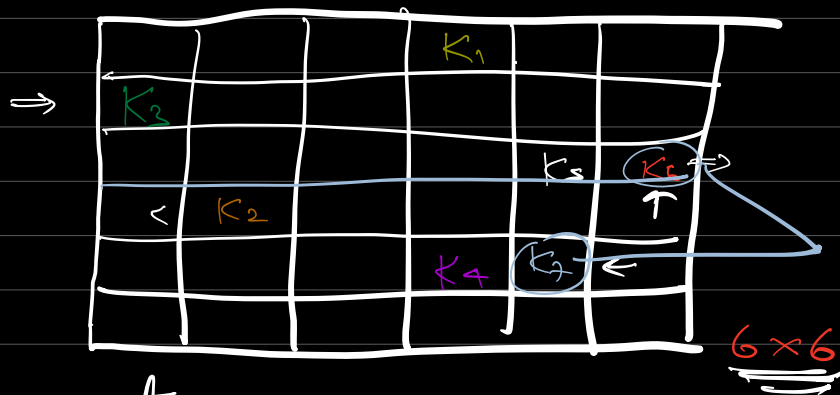
Q Given a chess board ($N \times N$).

You are also given K knights (Horse) placed on the board.

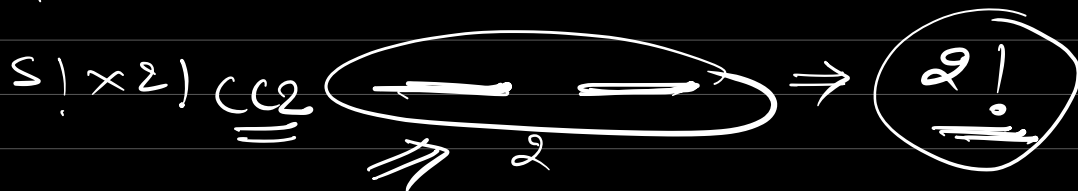
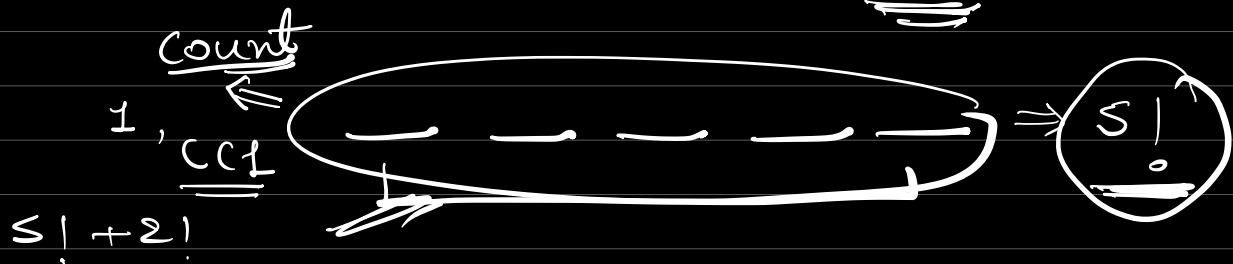
\Rightarrow If a knight is reachable from other knight. Then they can be swapped.

find no. of ways to arrange the knights

~~$K=5$~~ 7



Connected components.



Solⁿ

Find the count of elements of each cc.

& multiply the value of factorial of count.

Pseudo Code

ans = 1

```
for (i=0; i < N; i++) {  
    for (j=0; j < M; j++) {  
        if (M[i][j] == '1' & &  
            !visited(M[i][j])) {
```

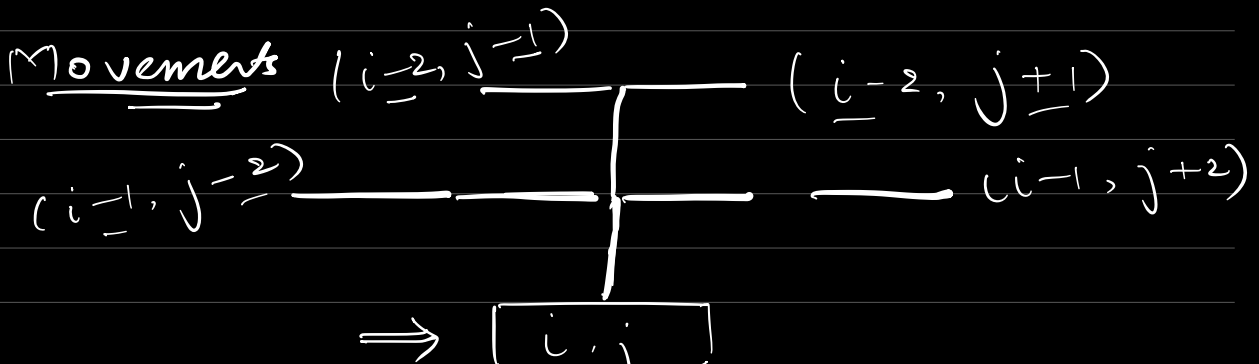
count = dfs(M[i][j]);

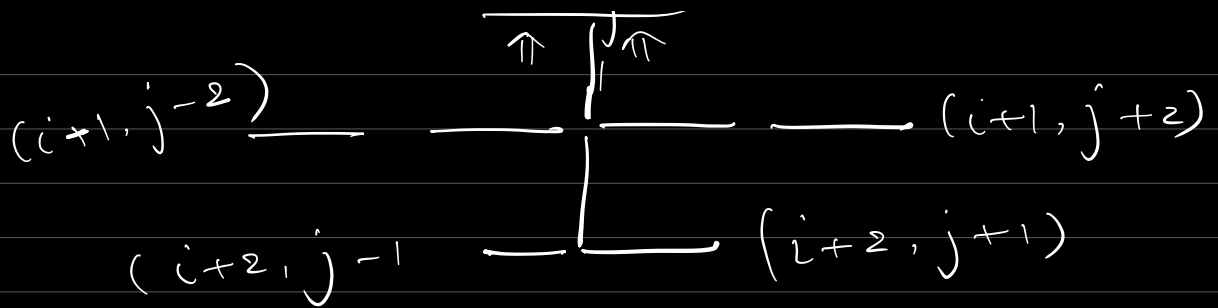
ans = ans * (count!)

}

}

}





Hack

Row Modifier $\Rightarrow -2, -2, -1, -1, +1, +1, +2, +2$
 Col Modifier $\Rightarrow -1, +1, -2, +2, -2, +2, -1, +1$

```
for (K=0; K < R.size(); K++) {
```

```
    temp-i = i + R[K];
```

```
    temp-j = j + C[K];
```

```
    check (temp-i, temp-j);
```

```
}
```

```
check (temp-i, temp-j) {
```

```
    if (i < 0 || j > n-1) {
        return;
    }
```

```
    if (j < 0 || i > m-1) {
```

' ~ set f_{in}

2
y set f_{in}