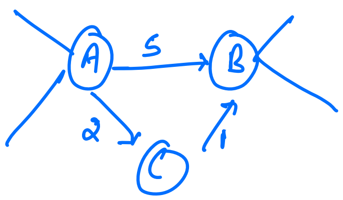
Good Evening Everyone !! ☺

Today's content

- Floyd Warshall
- Graph coloring
- Bi-partite Graph
- Construct Roads
- Rotten Oranges (very famous)

Q:) Find shortest distance from every node to every other node.

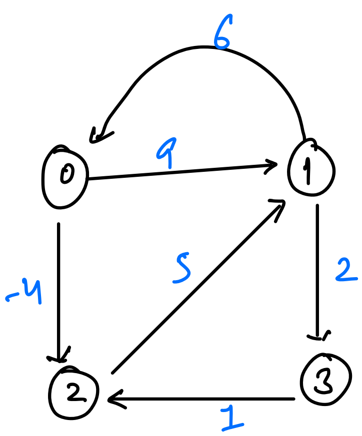
Floyd Warshall's Algorithm → All pair shortest path.



intermediate node

idea.→ Consider every node as intermediate node and try to relax the edges with larger edge wt.s.
(ignore)

Adjacency matrix



	0	1	2	3
0	0	9	-4	∞
1	6	0	∞	2
2	∞	5	0	∞
3	∞	∞	1	0

∞ → there is no edge from i to j.

	0	1	2	3
0	0	9	-4	∞
1	6	0	2	2
2	∞	5	0	∞
3	∞	∞	1	0

0 → intermediate node

$$\underbrace{d[u][0] + d[0][v] < d[u][v]}_{\text{update if}}$$

$d[1][0] + d[0][2] < d[1][2]$ ✓

$d[1][0] + d[0][3] < d[1][3]$ ✗

$d[2][0] + d[0][1] < d[2][1]$ ✗

$d[2][0] + d[0][3] < d[2][3]$ ✗

$d[3][0] + d[0][1] < d[3][1]$ ✗

$d[3][0] + d[0][2] < d[3][2]$ ✗

	0	1	2	3
0	0	9	-4	11
1	6	0	2	2
2	11	5	0	7
3	∞	∞	1	0

1 → intermediate node

	0	1	2	3
0	0	1	-4	3
1	6	0	2	2
2	11	5	0	7
3	12	6	1	0

2 → intermediate node

	0	1	2	3
0	0	1	-4	3
1	6	0	2	2
2	11	5	0	7
3	12	6	1	0

3 → intermediate node

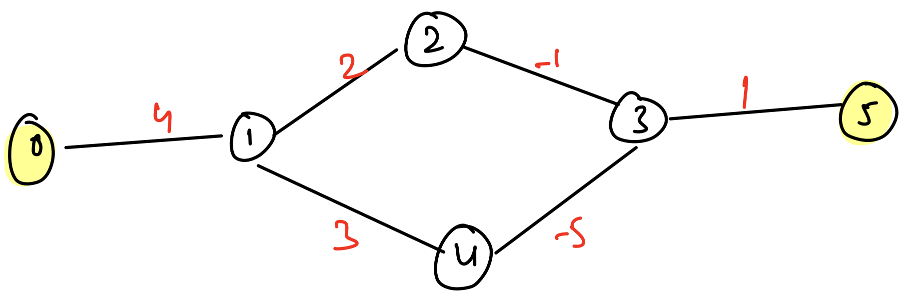
code.

```
for( K = 0 ; K < N; K++){
    for( i = 0; i < N; i++){
        for( j = 0; j < N; j++){
            if( d[i][K] + d[K][j] < d[i][j] ){
                d[i][j] = d[i][K] + d[K][j];
            }
        }
    }
}
```

T.C → $O(N^3)$
S.C → $O(1)$

Shortest distance b/w any 2 nodes is always possible?

No



$0 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 5$ (3)

$\underbrace{0 \rightarrow 1 \rightarrow 4 \rightarrow}\ \underbrace{3 \rightarrow 2 \rightarrow 1 \rightarrow 4}_{-1} \rightarrow \underbrace{3 \rightarrow 5}$ (2)

If -ve edge wt. cycle is present, then shortest dist. doesn't exist.

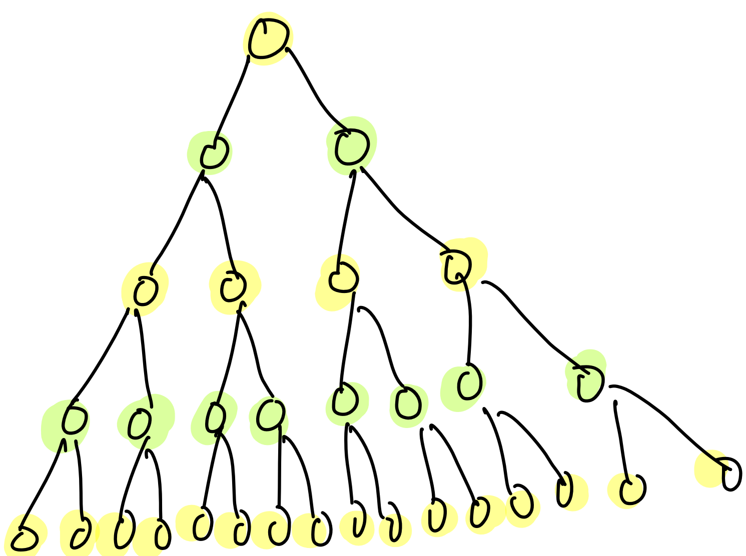
Graph Coloring.

Francis Guthrie. (1852)

Minimum no. of colors required to paint all the nodes in a graph such that no two adjacent nodes have the same color.

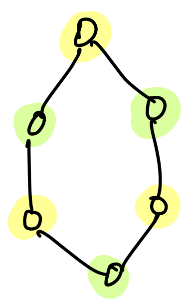
↳ **Chromatic Number**

① Tree

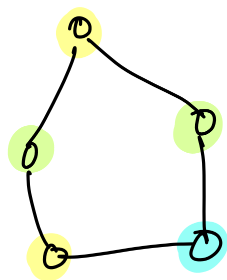


Chromatic no → 2

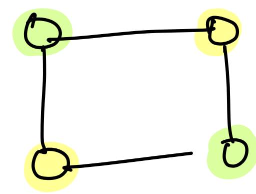
② Cycle Graph (whole graph is a cycle)



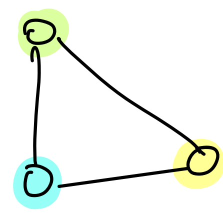
C.N = 2



C.N = 3



C.N = 2



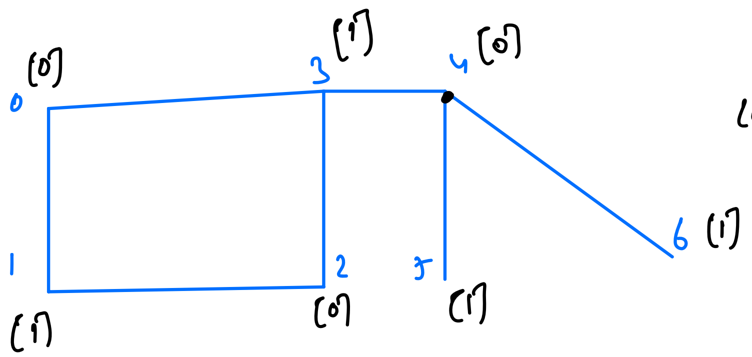
C.N = 3.

In general, C.N of cycle graph = $2 + (N \% 2)$

Bi-partite Graph

→ Any graph with chromatic number = 2. Eg → tree, even length cycle graph.

→ A graph is called bi-partite if we can divide all the nodes into two sets, such that all the edges are across the sets.

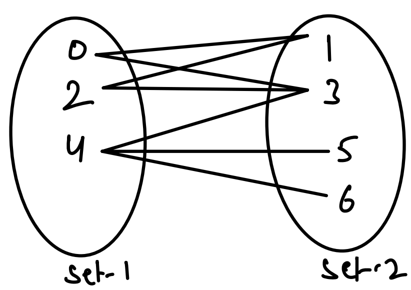


col[] →

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

0 → yellow

1 → green



Bi-partite Graph

Q. Check if the given graph is bi-partite or not?

check scaler day 92 Q->3 for better understanding

```
Col [N], ∀i, Col [i] = -1;
col [src] = 0;
boolean dfs( graph, src) {
    for( int nbr : graph (src]) {
        if( col(nbr] == col(src]) { return false }
        else if( col[nbr] == -1) {
            col[nbr] = 1 - col( src]; //opposite colour of src.
            if( dfs( graph, nbr) == false) {
                return false;
            }
        }
    }
    return true;
}
```

$$\boxed{\begin{array}{l} T.C \to O(N+E) \\ S.C \to O(N) \end{array}}$$

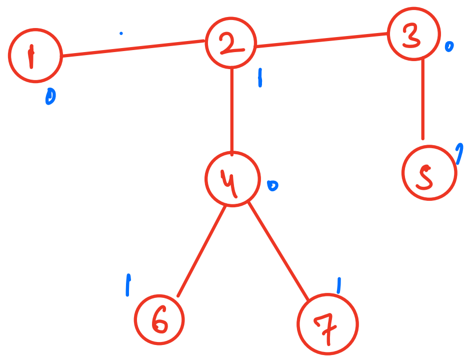
main f^n

```
for( i=0; i<N; i++) {
    if( col[i] == -1) {
        if( dfs( graph, i) == false) {
            return false;
        }
    }
}
return true;
```

checking for all the components.

Q.) A country consists of N cities connected by (N-1) roads. King of that country wants to construct maximum roads such that cities can be divided into two sets and there is no road between cities in the same set. Find maximum no. of new roads that can be created?

Note: All cities can be visited from any city.



[7 - cities
6 - roads]

Color →

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

Set → 1: 1, 3, 4 — Color → 0

Set → 2: 2, 6, 7, 5 — Color → 1

total possible roads = no. of nodes with color 0 * no. of nodes with color 1

$= 3 * 4 = 12.$

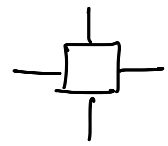
new roads that can be constructed $= 12 - 6 = 6.$

Rotten Oranges

Given a matrix [N*m] containing only 0's, 1's and 2's.

0 → empty cell, 1 → fresh orange, 2 → rotten orange.

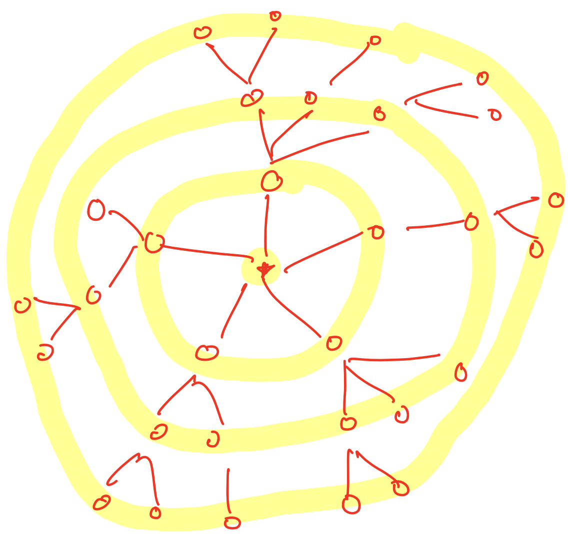
Every minute, all the fresh oranges adjacent to rotten oranges become rotten.



In how many time will all oranges become rotten?
If it is not possible, return -1.

	0	1	2	3	4
0	1	0	1	0	1
1	1	1	1	1	1
2	0	2	0	1	0
3	0	1	1	1	1
4	1	1	1	0	0

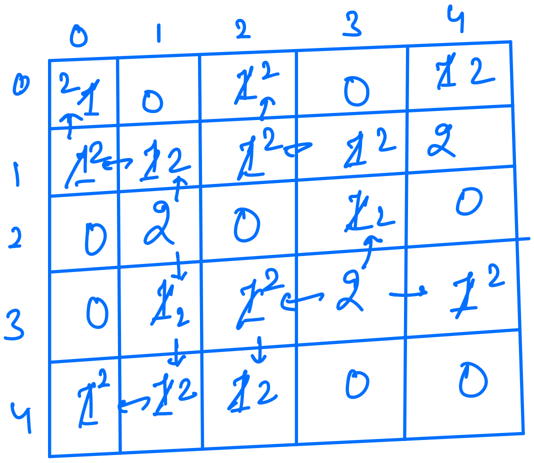
ans = 5.



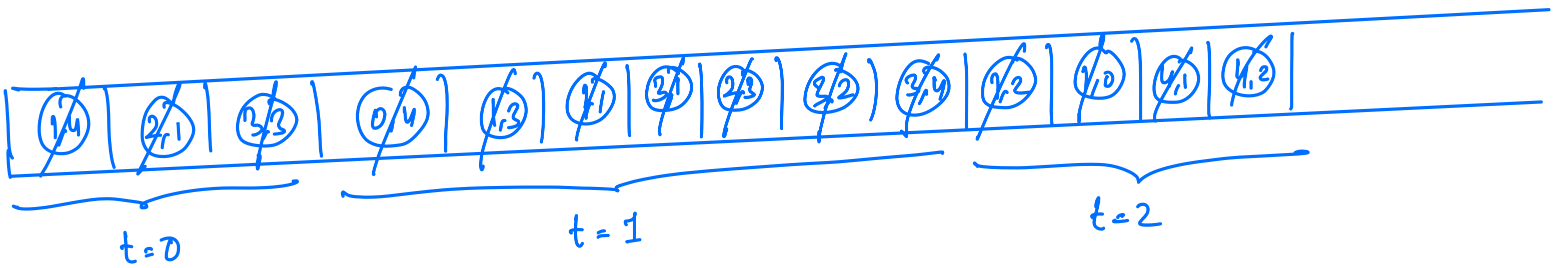
	0	1	2	3	4
0	1	0	1	0	0
1	1	1	1	0	0
2	0	2	0	0	1
3	0	1	1	0	0
4	1	1	1	0	0

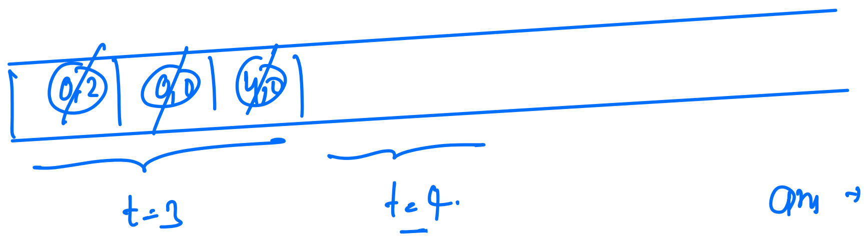
ans = -1

idea → B.F.S.



Multi-sourced B.F.S





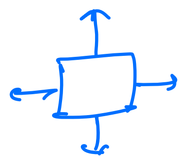
ans → t-1.

#code →

```
Queue<Pair> q;

for( i = 0; i < N; i++){
    for( j = 0; j < m; j++){
        if( arr[i][j] == 2){ q.enqueue(new Pair(i,j)); }
    }
}

T = 0;
while( q.isEmpty() == false){
    sz = q.size();
    for( i = 1; i <= sz; i++){
        Pair rp = q.dequeue();
        if( i-1 >= 0  &&  arr[i-1][j] == 1){
            arr[i-1][j] = 2;
            q.enqueue(new Pair(i-1,j));
        }
        if( j-1 >= 0  &&  arr[i][j-1] == 1){
            arr[i][j-1] = 2;
            q.enqueue(new Pair(i,j-1));
        }
        if( i+1 < N  &&  arr[i+1][j] == 1){
            arr[i+1][j] = 2;
            q.enqueue(new Pair(i+1,j));
        }
```



```
        if (j+1 < m && arr[i][j+1] == 1) {
            arr[i][j+1] = 2;
            q.enqueue(new Pair(i, j+1));
        }
    }
    T++;
}

for( i=0; i<n; i++) {
    for( j=0; j<m; j++) {
        if( arr[i][j] == 1) {
            return -1
        }
    }
}
return T-1;
```

$$\begin{bmatrix} T.C \rightarrow O(N*m) \\ S.C \rightarrow O(N*m) \end{bmatrix}$$

---

1. PSP ≥ 75%
2. Contest → 15 Dec. [D.P, Graphs]
3. Revision
4. Mock Interview ⇒ Schedule by 30th Dec
5. Actual Interview from D.S.A side.