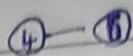
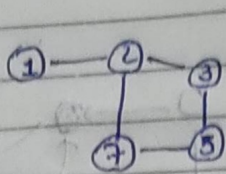


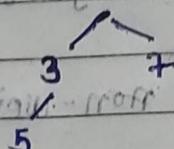
* BFS (Breath First Search)



→

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

→ start from 1 → 2



Take visited array
for checking

→ start from 4 → 6

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

when we visit element
we make them true

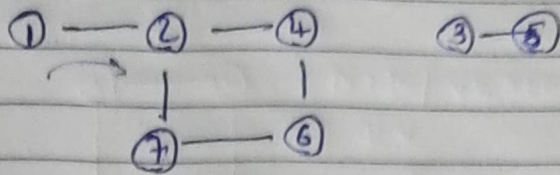
→ TC → $O(N + E)$, $N = \text{nodes}$
 $E = \text{travelling through}$

① — ② ③ — ④ adjacent nodes

→ SC → $O(N + E) + O(N) + O(N)$

↳ For Adj list, array and queue

* DFS (Depth First Search)



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

→ output → 1 → 2 → 4 → 6 → 7 → 3 → 5

0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8

visited array

Mark true when we visit element.

→ For (i = 1 → n)

{

if (vis[i] == false)

{ DFS(i) }

}

DFS(1)

DFS(2)

DFS(4)

DFS(6)

DFS(7)

DFS(3)

DFS(5)

DFS(3)

DFS(5)

→ TC and SC same as BFS.