GRAPHS...

video-36

"let's make it easy too"

Clean, elegant-

HARD

Leetcode -847

If you have tried my Graph Concepts & Qns playlist, these Qns, will seem very easy. Do try it once :)

Shortest Path Visiting All Nodes

Company tags :- Google

BFS

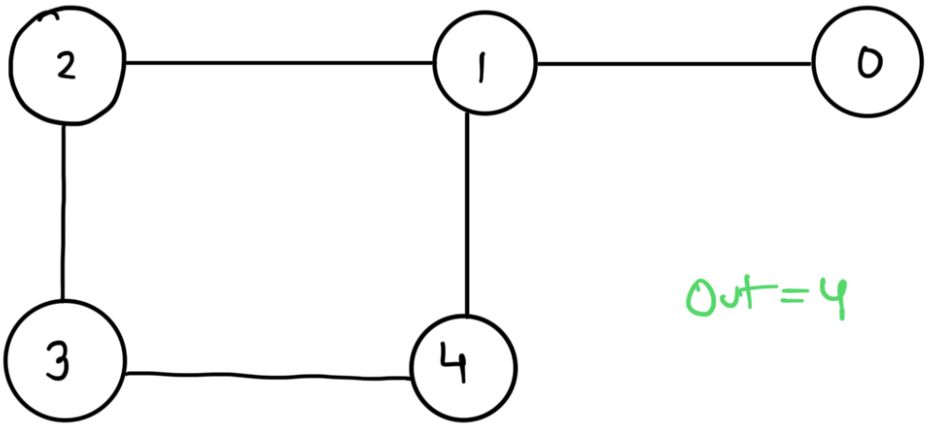
847. Shortest Path Visiting All Nodes

Hard 3538 147 Add to List Share

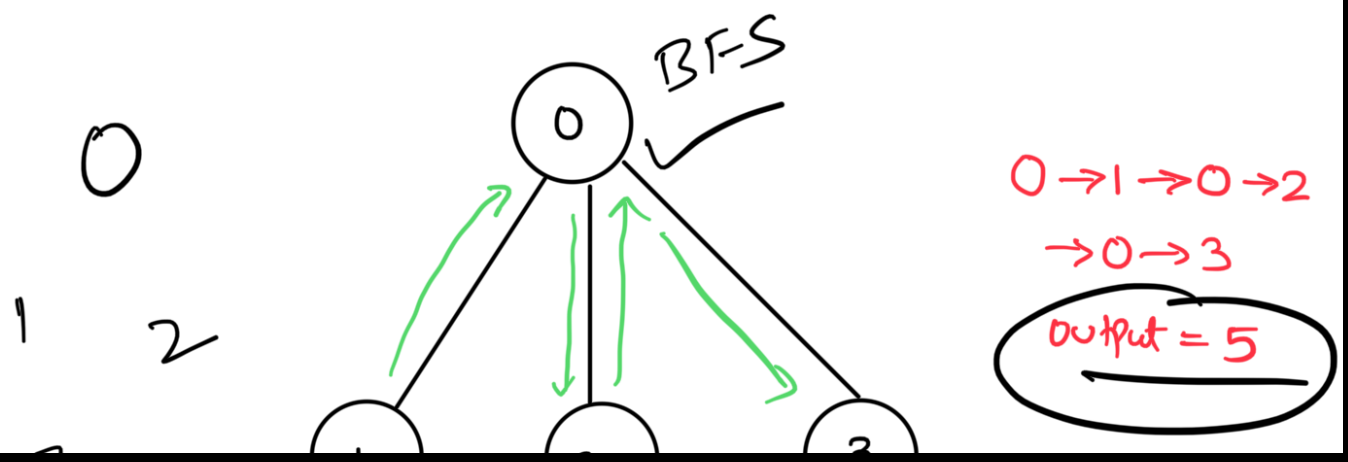
You have an undirected, connected graph of `n` nodes labeled from `0` to `n - 1`. You are given an array `graph` where `graph[i]` is a list of all the nodes connected with node `i` by an edge.

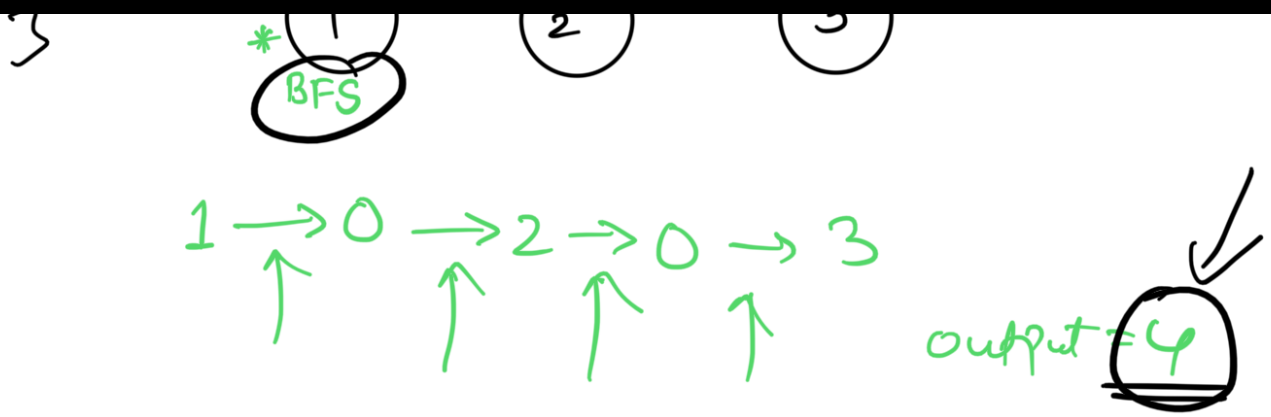
Return *the length of the shortest path that visits every node*. You may start and stop at any node, you may revisit nodes multiple times, and you may reuse edges.

Example :



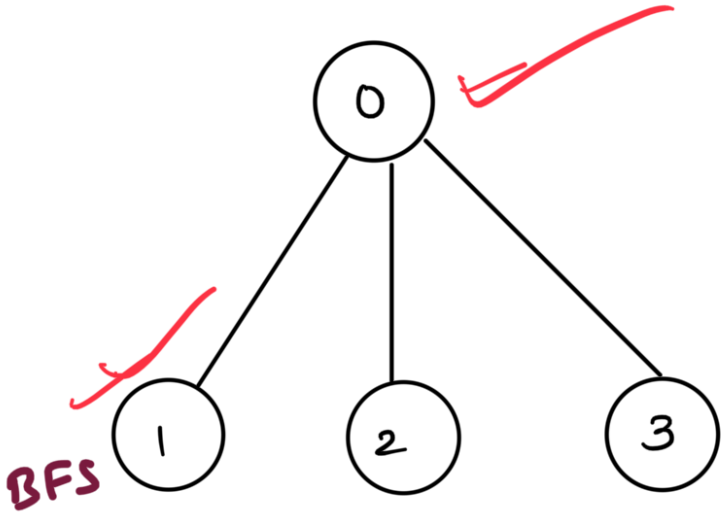
Out = 4

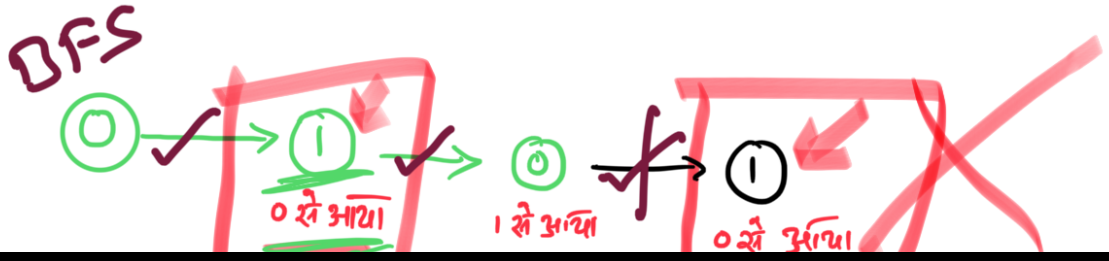


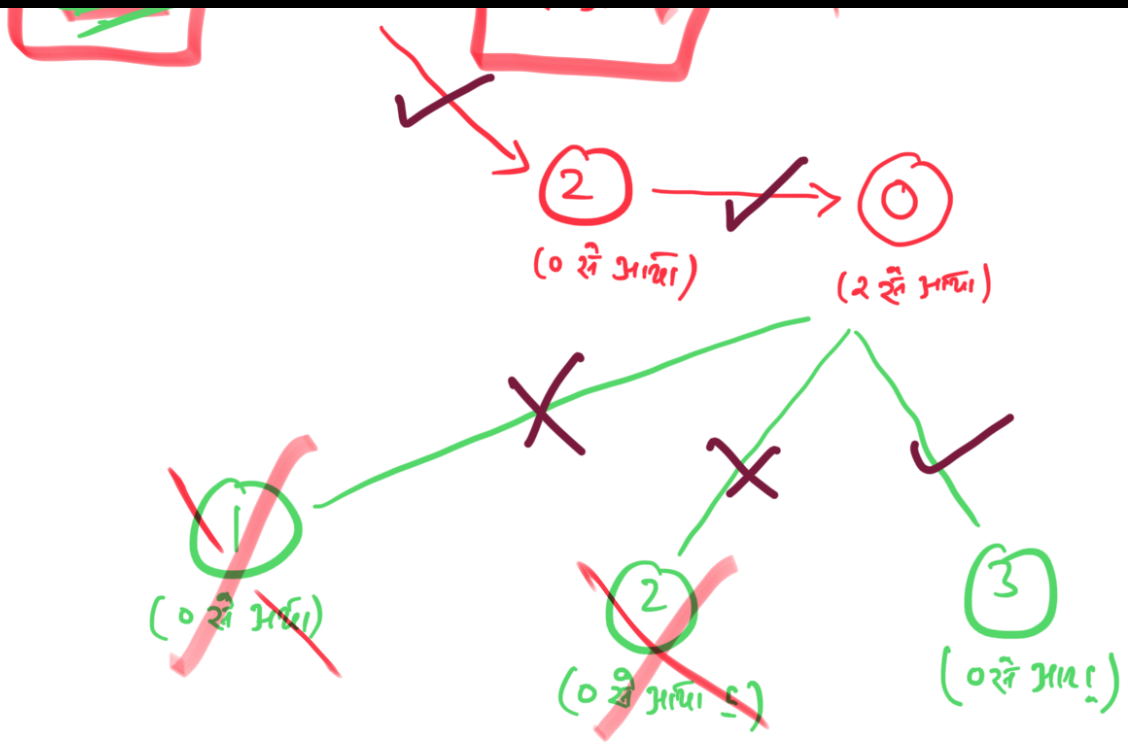


queue.

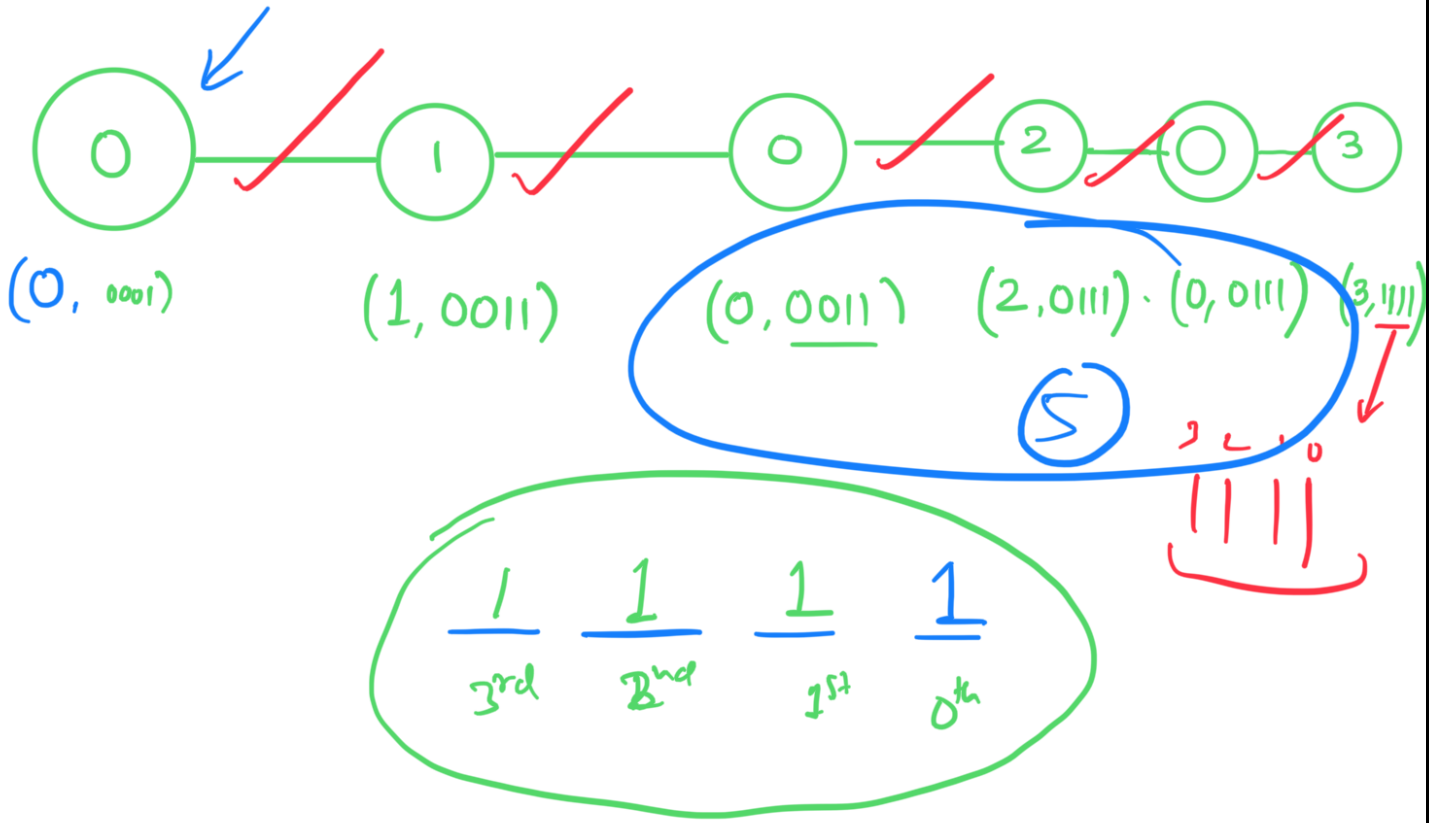
① Try BFS from all possible nodes.

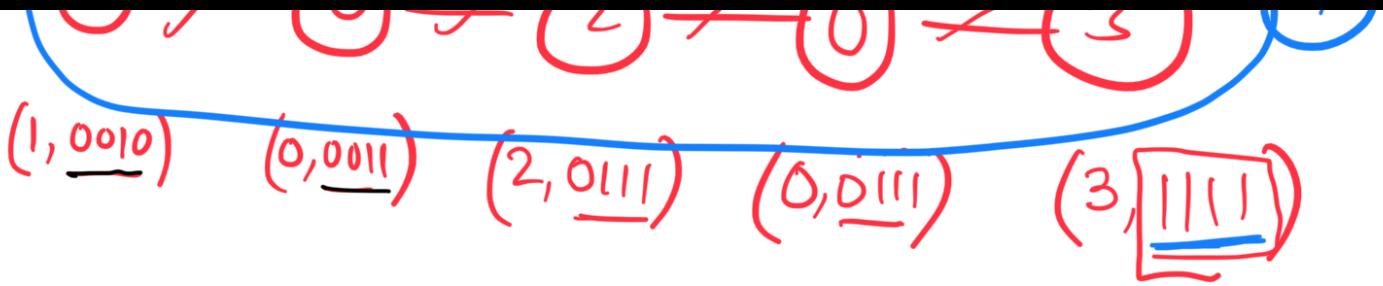





2
(0 से आया)
0
(2 से आया)
1
(0 से आया)
2
(0 से आया)
3
(0 से आया)

Path = 5


0
1
0
2
0
3
(0, 0001)
(1, 0011)
(0, 0011)
(2, 0111)
(0, 0111)
(3, 1111)
5
1 1 1 1
3rd 2nd 1st 0th



return path.

Queue (BFS)

{0, 0001} 1	{1, 0010} 2	{2, 0100} 4	{3, 1000} 8

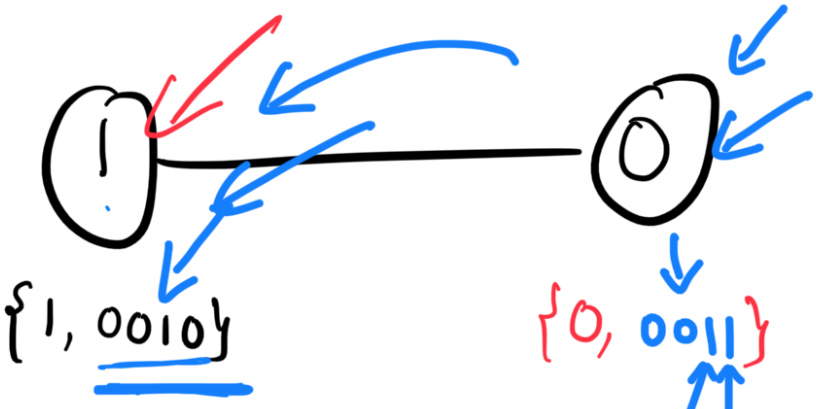
F.S = $(2^n - 1)$ → 1111

node mask

(•) C++ (Set <pair <int, int>> visited;

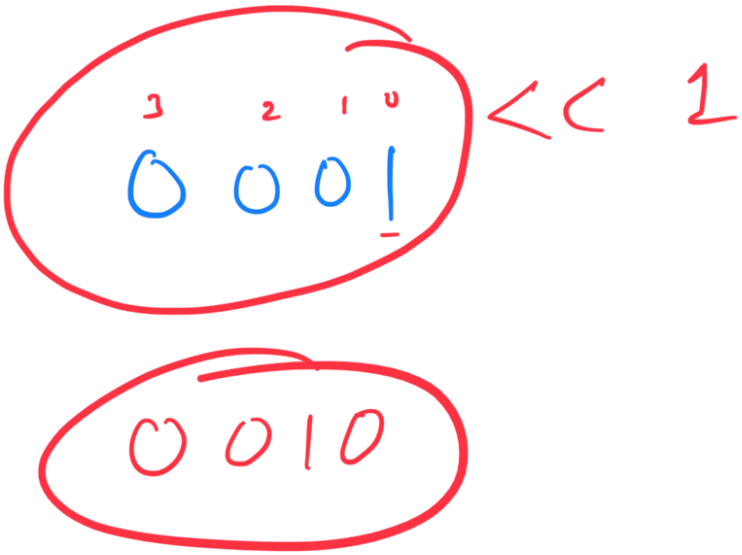
(•) Java ([n] [$(2^n-1)+1$] boolean

Bit.



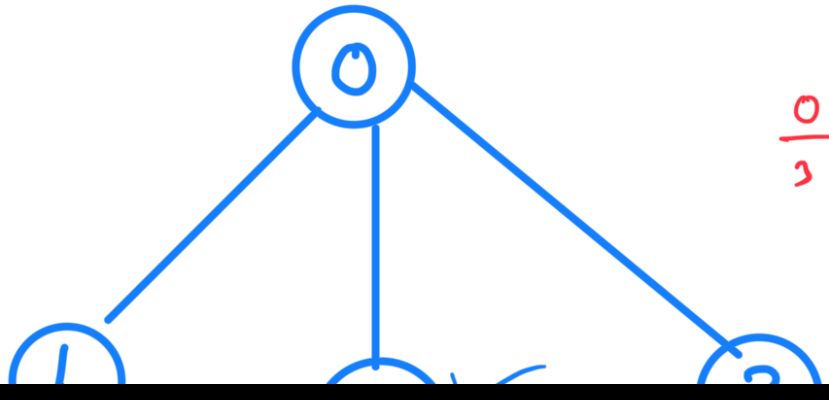
0 0 1 0
0 0 0 1
0 0 1 1

0001 << 0



Full Dry Run :-

0 0 0 1



0/3 1/2 0/1 0/0

Paths = 1 + 1

node = 0
mask = 5 (0101)

next = 3
nextmask = 1000 OR 0101 = 1101

queue {3,13}

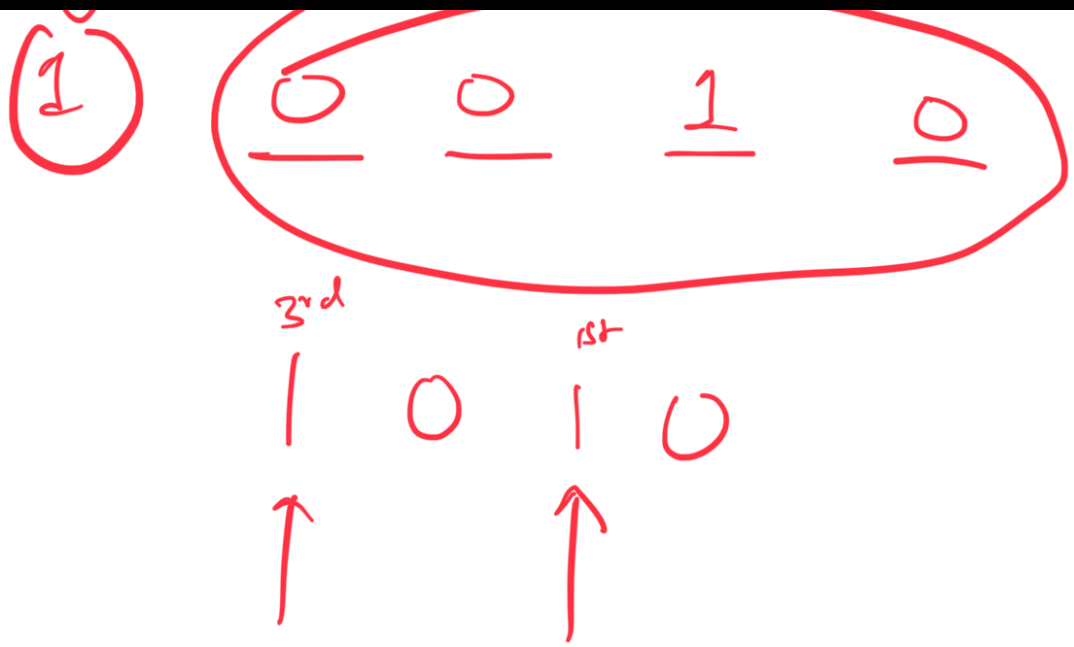
{1,3} | {0,3} | {0,5} | {0,9} (27) (13)

visited

{0,1} {2,4} {1,2} {3,8} {0,3} {1,3} {0,5} {0,9} {2,7} {3,13}

1. BFS from all nodes
2. visited { node, state }
3. que { node, state }

3 → 1 0 0 0 (Bit)


1
0 0 1 0
3rd
1st
1 0 1 0