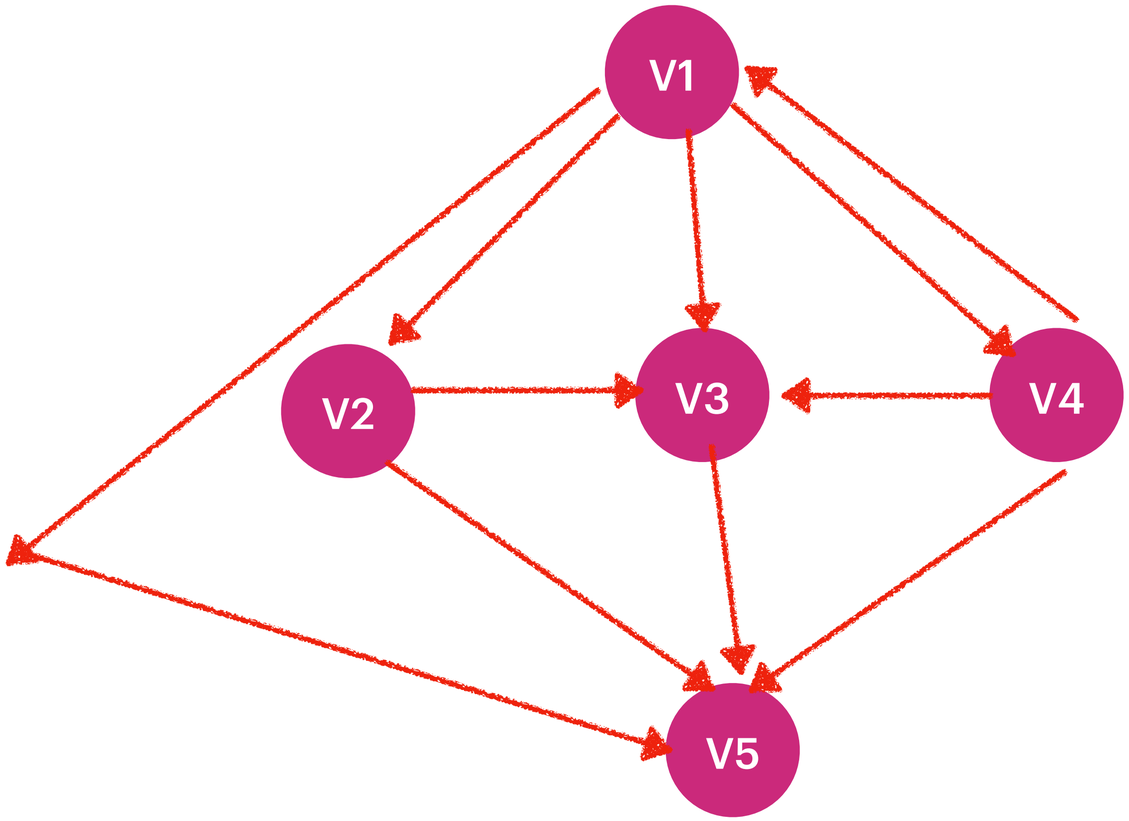


V * 2 ^ V

In worst case as per Graph Theory For each vertex we can make (2^V-1 - 1) unique paths. So in worst cast time complexity is O(V*2^V)

Source (V1) Target (V5)

V1 -—> V5 through V3 IFF V1——>V3 & V3——>V5

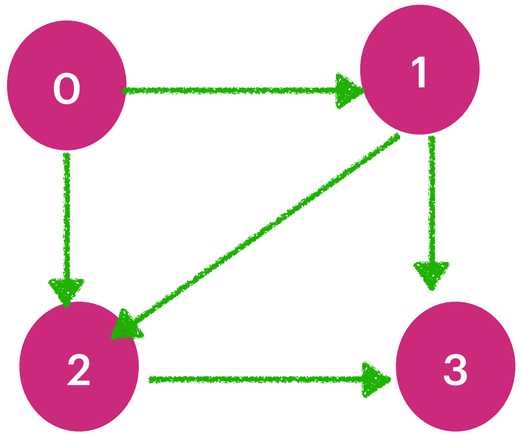
V1 -—> V5 through V2 IFF V1->V2 & V2->V5

All Paths From Source to Target : DFS

Input: graph = [[1,2],[2,3],[3],[]] n=4
Output: [[0,1,3],[0,2,3],[0,1,2,3]]
Explanation: There are two paths: 0 -> 1 -> 3 and 0 -> 2 -> 3.
0 -> 1 -> 2 ->3

0 1 2 3

graph = [[1,2], [2,3], [3], []] n=4



Empty

Pop [0,1,2,3]
Last element in the path is target

[0,1,2,3]

Pop [0,1,2] => Added edges of 2 to the path
[0,1,2,3] push to stack

[0,1,2]

Pop [0,1,3]
Last element in the path is target, take it

[0,1,3]
[0,1,2]

Pop [0,1] => Added edges of 1 to the path
[0,1,2] [0,1,3] push to stack

[0,1]

[0,2,3]
[0,1]

Pop [0,2,3]
Last element in the path is target, take it

[0,2]
[0,1]

Pop [0,2] => Added edges of 2 to the path
[0,2,3] push to stack

[0]

[0] -> 1,2

Paths From 0 to 3

0 —> 1 —> 3 , 0—> 2 —> 3 ,
0 —> 1 —> 2 —> 3