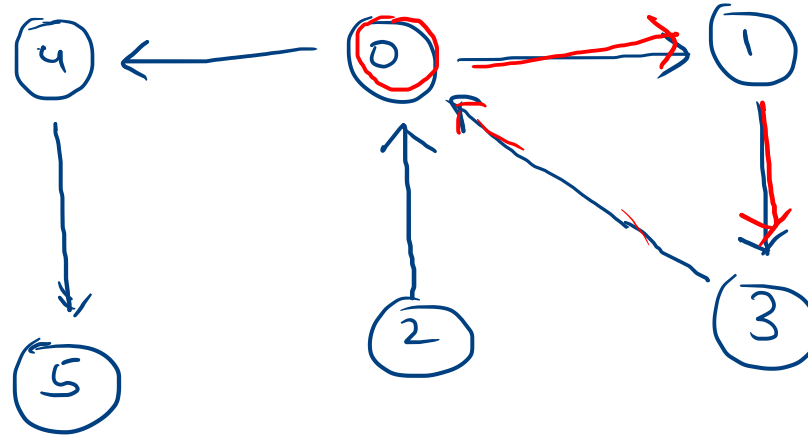


# Detect Cycle in Directed Graph

G =

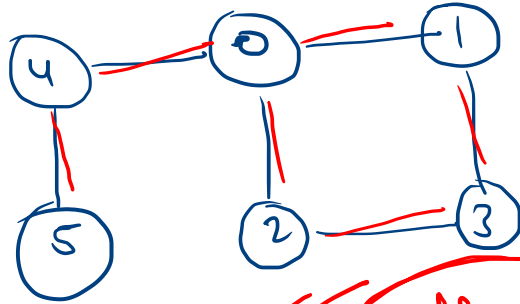


0-1-3  
Cycle

Let's suppose

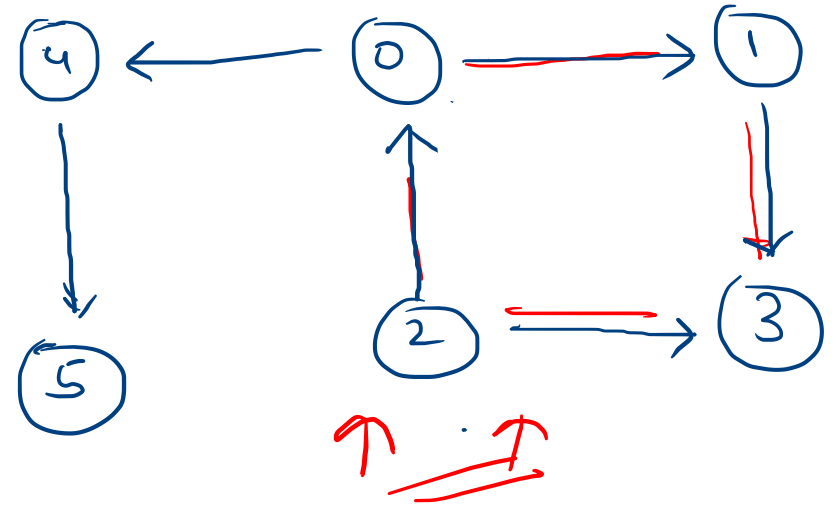
It is undirected Graph

→



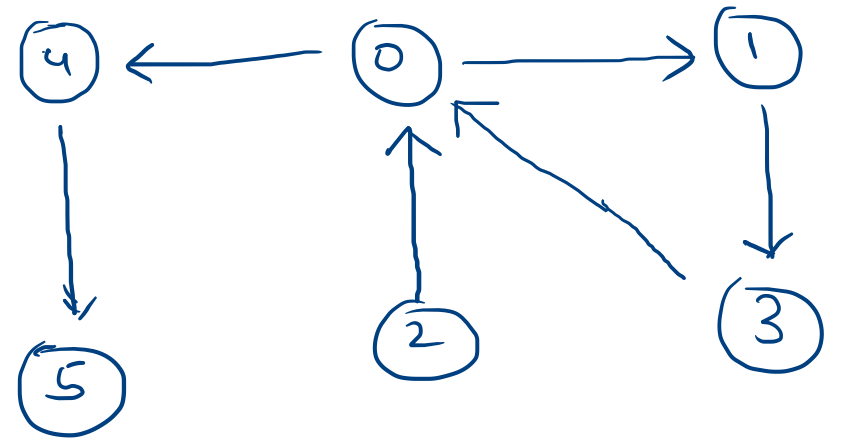
Cycle  
0-1-3-2

$G \Rightarrow$



Logic → maintain, visited  
dfs visited

- ① create visited, dfs visited, boolean vector
- ② for (every vertex u)  
if ( ! visited[u] )  
< if ( cyclic ( u, visited, dfs visited ) )  
return true;  
>  
return false



- True
- cyclic ( u, visited, dfs visited )
- < ① mark u to true in both vector
  - ② for (every adjacent vertex v of u )  
if ( not visited )  
call cyclic ( v, visited, dfs visited )  
else if ( visited[v] == true & & dfs visited[v] == true )  
return true
  - ③ dfs visited[u] = false ;  
return false;

visited

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

~~dfs visited~~

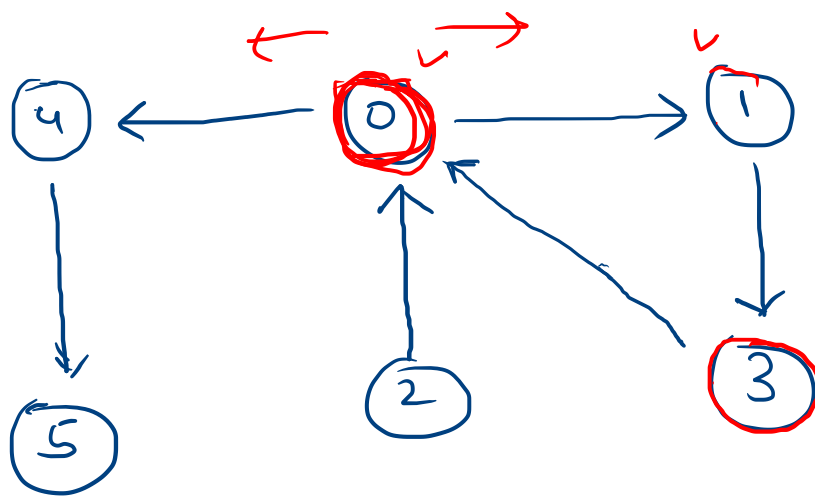
~~current visited~~

DFS(0) → cyclic(0) → cyclic(1)

DFS(0) → cyclic(4)

cyclic(2)

True



0 → 1 4  
1 → 3  
2 → 0  
3 → 0  
4 → 5  
5 →

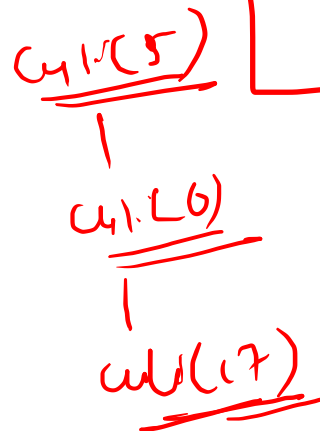
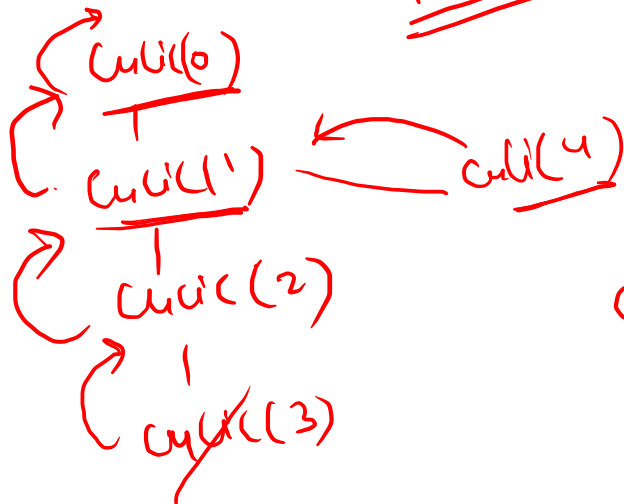
for i = 0; i < n; i++  
 if (!visited[i])  
dfs

visited

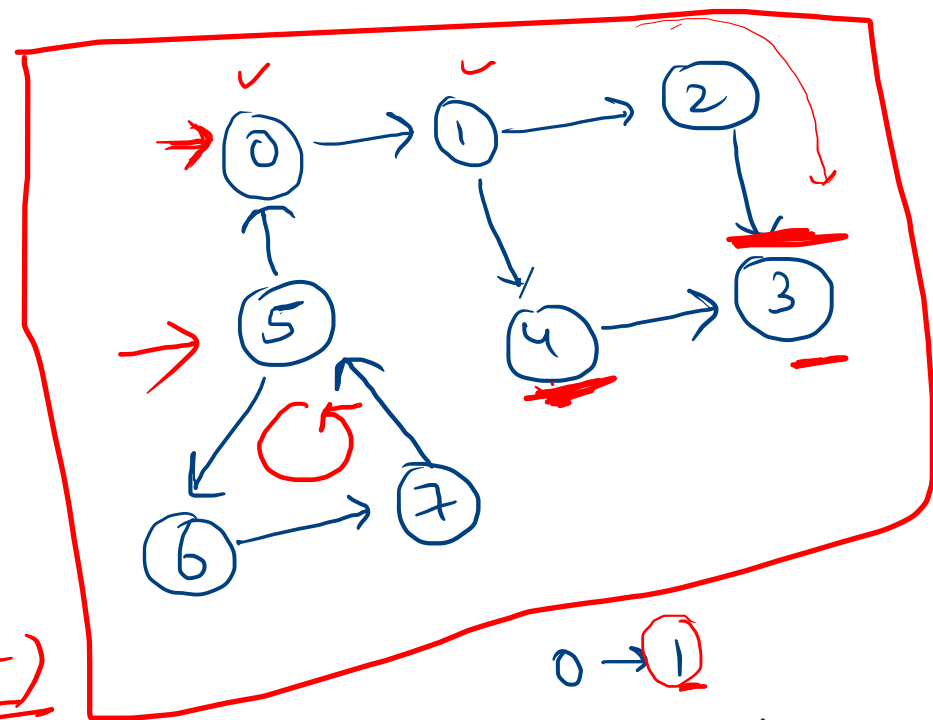
dfs visited

0	1	2	3	4	5	6	7
<del>1</del>	<del>1</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>
<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>

true



( )



0 → 1  
 1 → 2, 4  
 2 → 3  
 3 → -  
 4 → 3  
 5 → 0, 6  
 6 → 7  
 7 → 5