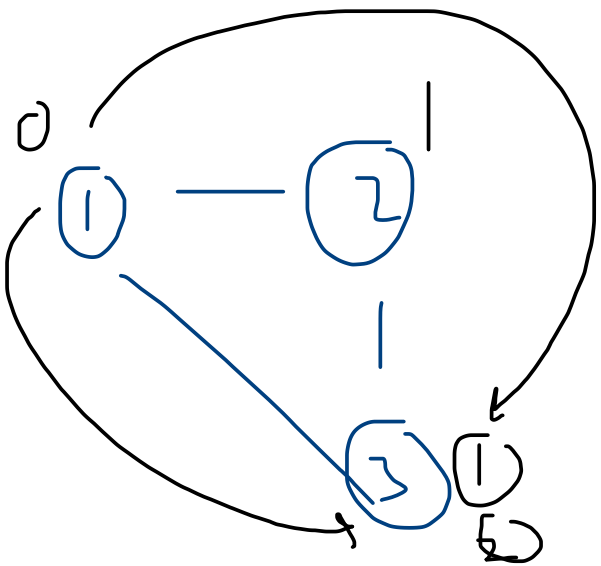
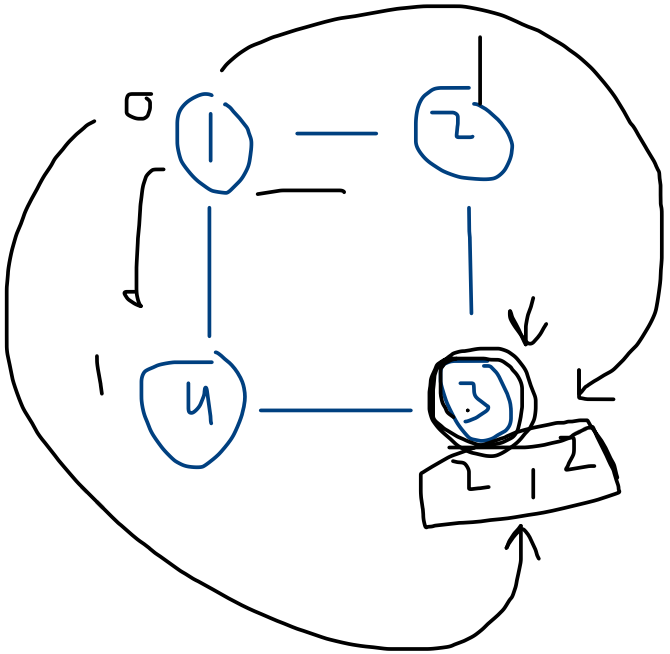
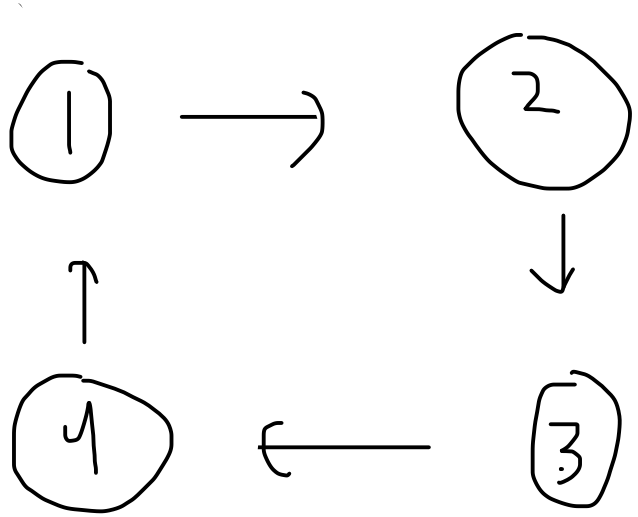
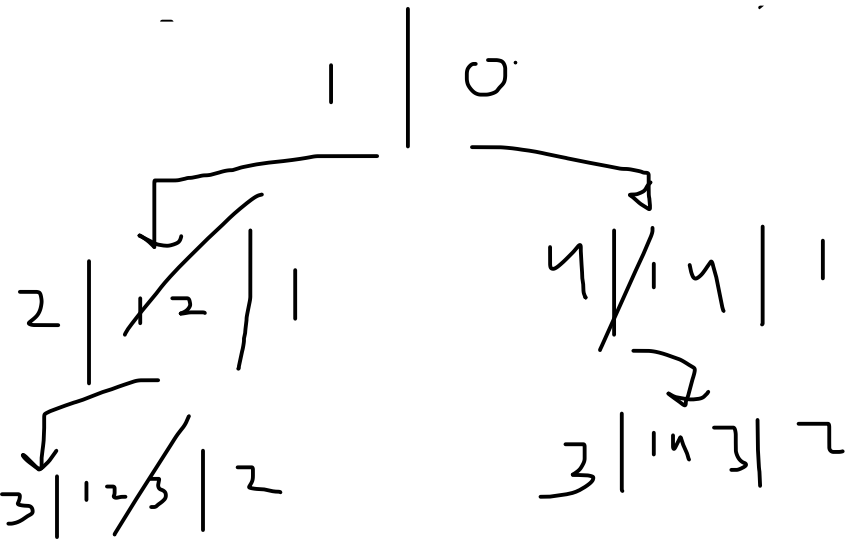
isBipartite

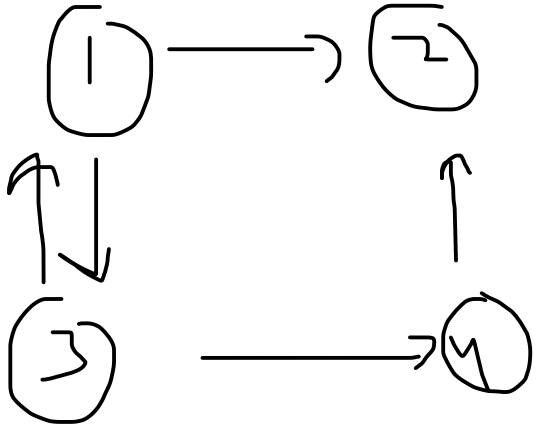


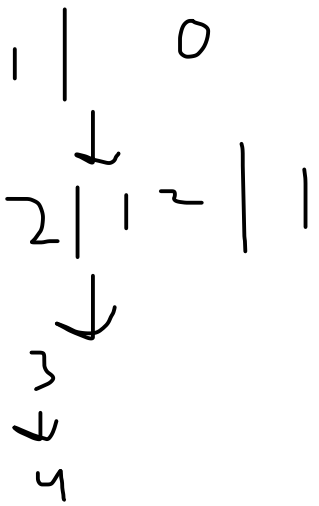


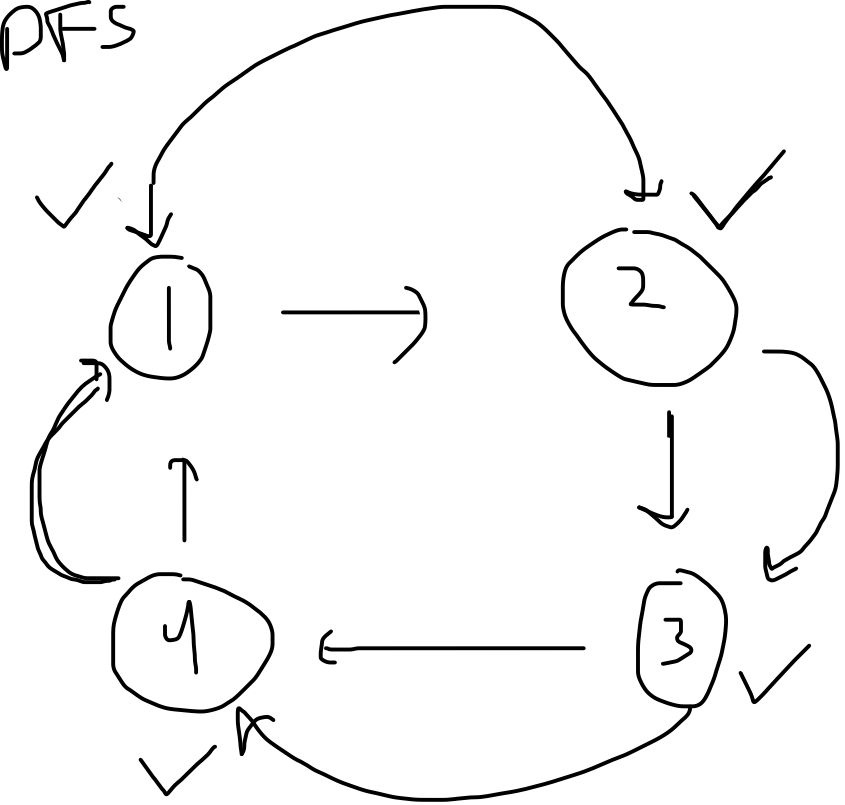
Detect Cycle in a Directed Graph

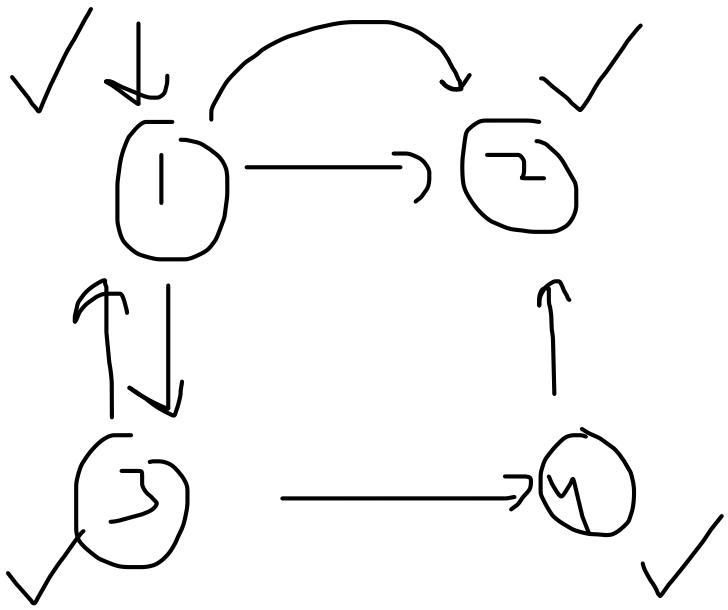


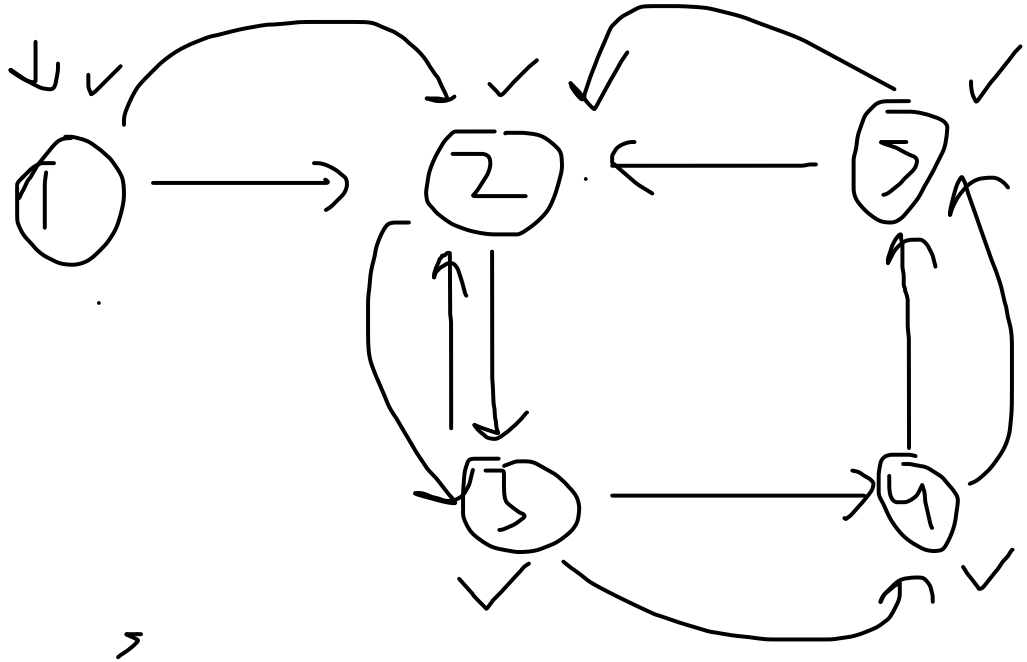


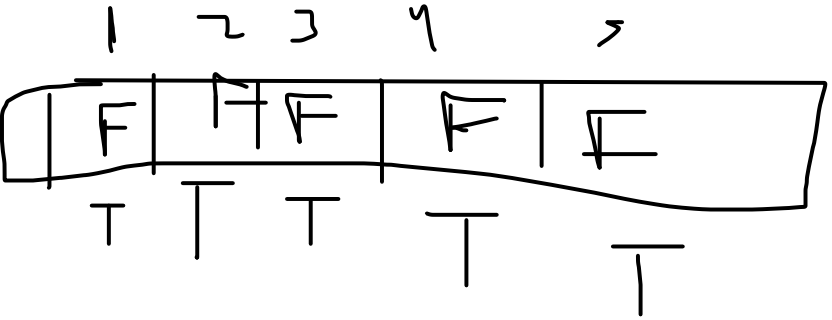


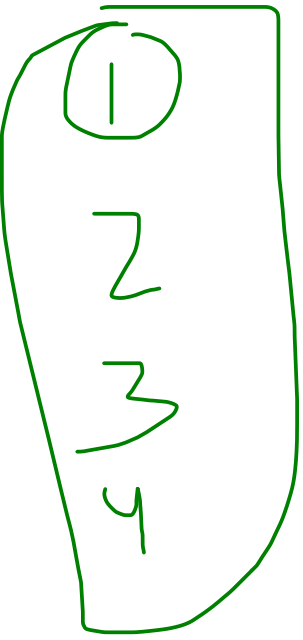


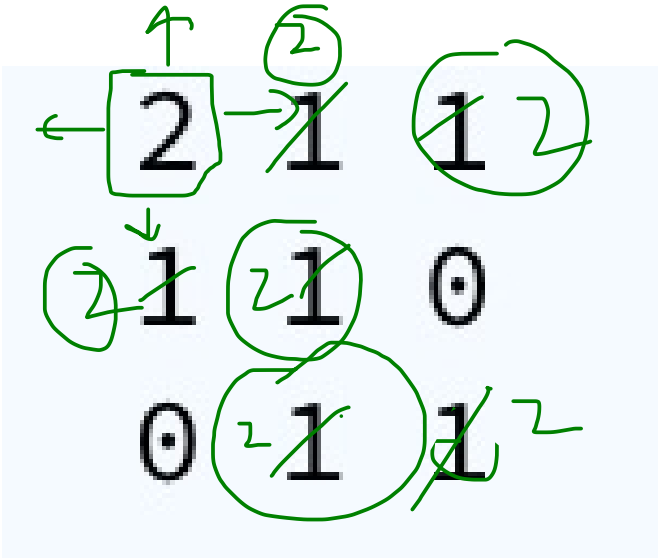

DFS
1
2
4
3

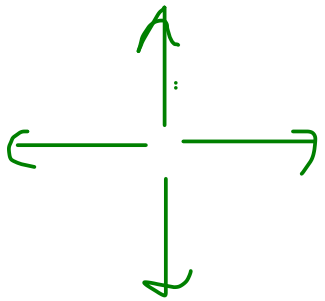

1
2
3
4

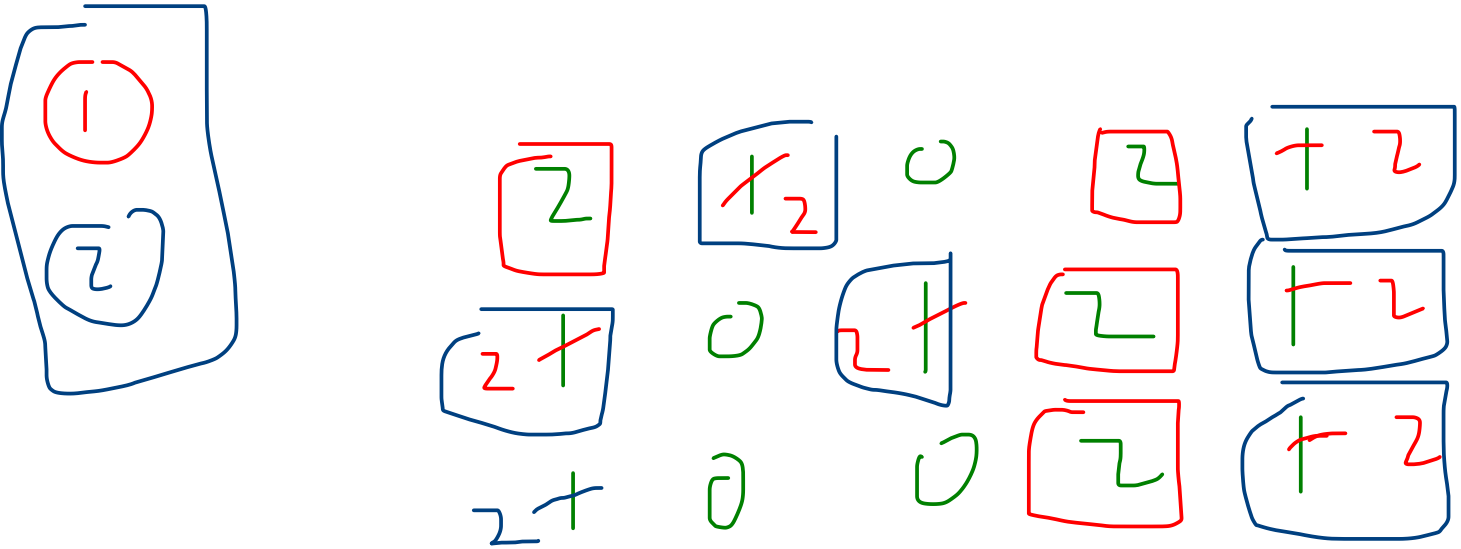

1
2
3
4
5


1
2
3
4
5
F
F
F
F
T
T
T
T
T

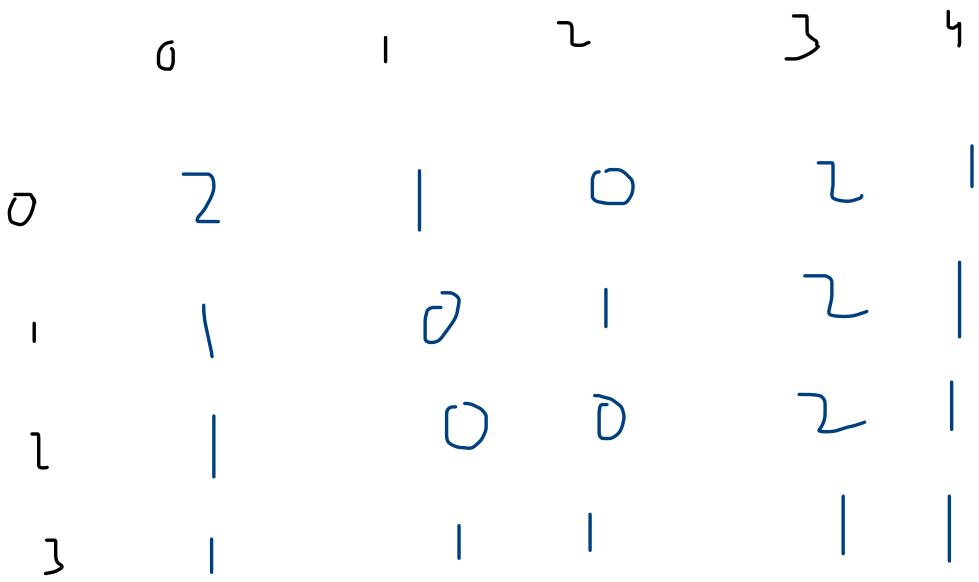


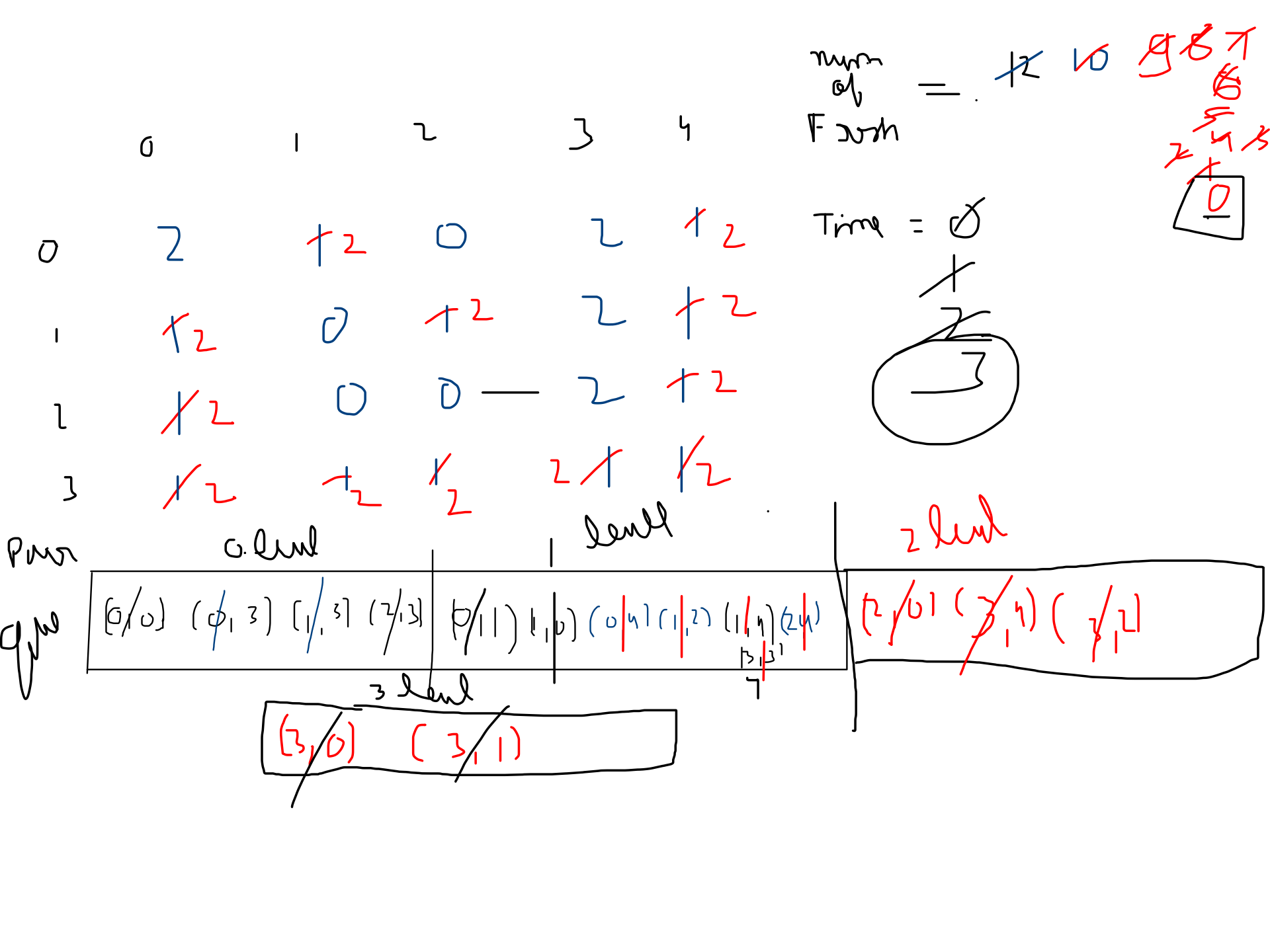

2 1 1
1 1 0
0 1 1

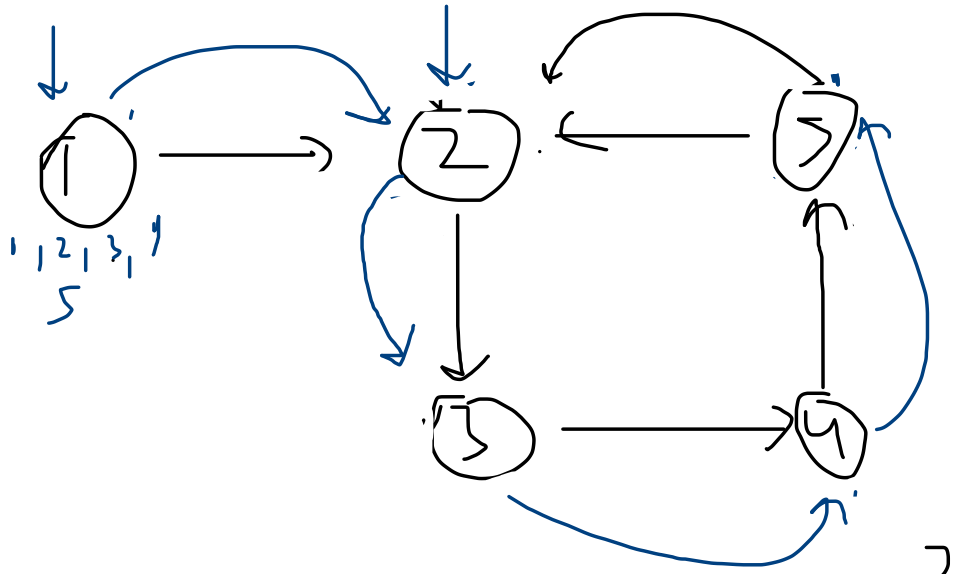




multisource BFS




num of Fresh
0 1 2 3 4
Time =
0 level
1 level
2 level
3 level
Prev
Queue



```java
// base cases
if (inStack[vtx]) {
    return true;
}
if (visited[vtx]) {
    return false;
}
visited[vtx] = true;
inStack[vtx] = true;
for (Edge e: graph[vtx]) {
    if (isCycle(e.nbr, visited, inStack, graph)) {
        return true;
    }
}
inStack[vtx] = false;
return false;
```