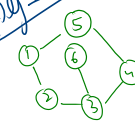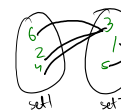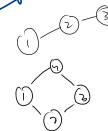# is Bipartite

Def 1

Theoretical definition:

If we are able to divide vertices of a graph into 2 mutually exclusive and exhaustive set, such that all edges are across set, the graph is said to be bipartite.
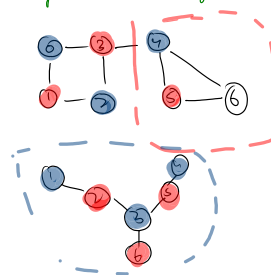
set1   set2

```java
public boolean bfs(ArrayList<ArrayList<Integer>> graph, int[] visited, int src){
    Queue<Pair> q = new LinkedList<>();
    q.add(new Pair(src,1));
    while(q.size()!=0){
        Pair curr = q.remove();
        //mark
        if(visited[curr.node] != 0) continue;
        visited[curr.node] = curr.color;
        int oppColor = curr.color == 1?-1:1;
        //add nbrs
        for(int nbr:graph.get(curr.node)){
            if(visited[nbr] == 0){
                //nbr is colorless and i can color it with oppColor
                q.add(new Pair(nbr,oppColor));
            }else if(visited[nbr] == curr.color) return false;
            //the nbr is already having oppcolor, i will continue
        }
    }
    return true;
}
```

Vis = | 1 | -1 | 1 | -1 | 1 |
      | 0 | 1 | 2 | 3 | 4 |

{4, 1}

{4, -1}

Def 2

Acyclic => Bipartite

Cyclic => Even cycle ✓
          Odd Cycle ✗

1   2

Def 3
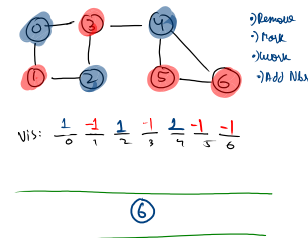
If I am able to color the nodes of graph with 2 colors such that neighbours are not having the same color

DFS / BFS

1 => Blue
0 => colorless
-1 => red

1)Remove
2) Mark
3) Queue
4) Add Nbr

Vis: | 1 | -1 | 1 | -1 | 1 | -1 | 1 |
     | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

(6)

6

Topological sorted order

DAG

1 2 6 5 3 4

integr => | 0 | 1 | | | |
          | 1 | 2 | 3 | 4 | 5 |

M having indeg=0

1)Remove
2)Mark
3)work
4)add Nbr

[3 1]
[5 1]

2 6 1 3 5 4

Kahn's algo

i) Calculate indegrees
ii) Add indegree=0 into Q
iii) start BFS

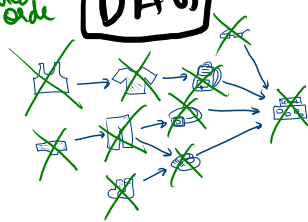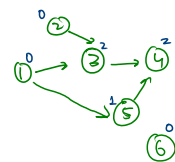## Kahn's algo

i) Calculate dependencies
ii) while(sm1 unvisited)
    choose 0 depen
    do work
    reduce dep of its nbr

2 6 1 3 5 4