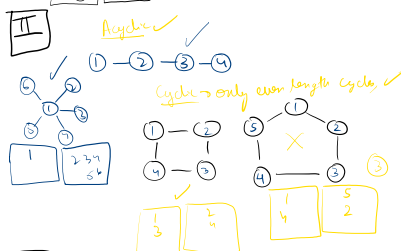
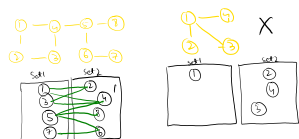
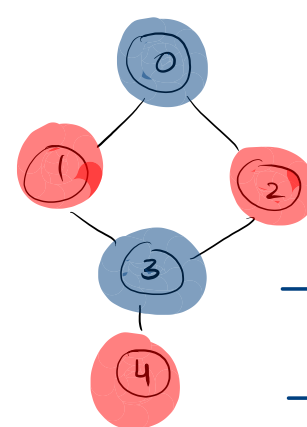
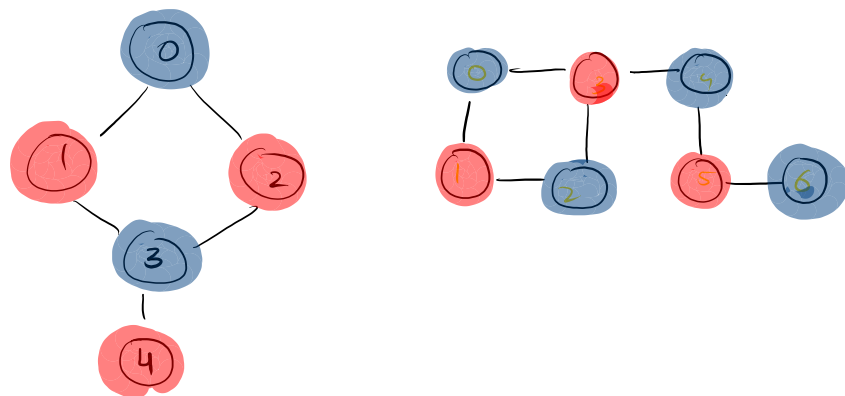


I Bipartite



III If you can color the nodes of graph with 2 colors such that nodes are having different color graph is bipartite.



1 \Rightarrow Blue
-1 \Rightarrow Red
vis: $\frac{1}{0}$ $\frac{-1}{1}$ $\frac{-1}{2}$ $\frac{1}{3}$ $\frac{-1}{4}$

* Remove
* Mark
* ~~visit~~
* Add unvisited nbs

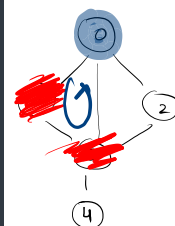
```
public boolean isBipartite(ArrayList<ArrayList<Integer>> graph) {
    Queue<Pair> q = new LinkedList<>();
    q.add(new Pair(0, 1)); // I'll try to color 1 with color

    while(q.size() > 0) {
        Pair curr = q.remove();

        if(vis[curr.node] != 0) continue;
        vis[curr.node] = curr.color;

        int oppColor = curr.color == 1 ? -1 : 1;

        for(int nbr: graph.get(curr.node)) {
            if(vis[nbr] == 0) {
                q.add(new Pair(nbr, oppColor));
            } else if(vis[nbr] == curr.color) return false;
        }
    }
    return true;
}
```



1 \Rightarrow Blue
-1 \Rightarrow Red
vis: $\frac{1}{0}$ $\frac{-1}{1}$ $\frac{-1}{2}$ $\frac{1}{3}$ $\frac{-1}{4}$

* Remove
* Mark
* ~~visit~~
* Add unvisited nbs