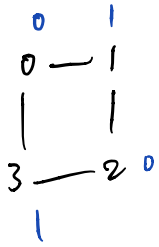
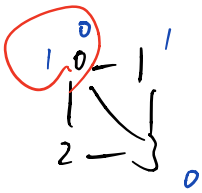
[[1,3],[0,2],[1,3],[0,2]]



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

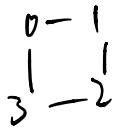



```
const fn = (graph, v = 0, label = 0, visited = {}) => {
    if ( v in visited) {
        return label === visited[v];
    }
    visited[v] = label;
    for (const u in graph[v]) {
        if ( ! fn(graph, u, (label + 1) % 2, visited)) {
            return false;
        }
    }
    return true;
};
```

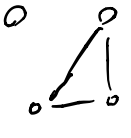
of

fail ⇒ didn't handle disconnected components!

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



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



⇒ is bipartite

```
const isBipartite = (graph) => {
    Conse n = graph.length;
    Conse visited = {};
    for(let i=0; i<n; i++){
        if(!(i in visited)){
            if(!dfs(graph, i, 0, visited)){
                return false;
            }
        }
    }
    return true;
}

function dfs(graph, u, label, visited){
    if(u in visited){
        return label === visited[u];
    }
    visited[u] = label;
```

```
    for(const v of graph[u]) {
        if(!dfs(graph, v, (label+1)%2, visited)) {
            return false;
        }
    }
    return true;
}
```