684. Redundant Connection

0, 1

1, 4

5, 2

2, 6

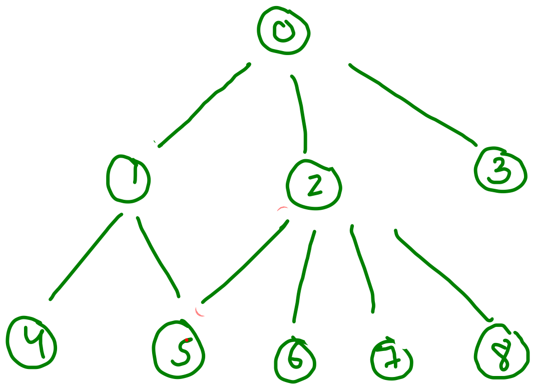
0, 2

6, 7

$\boxed{1, 5}$

0, 3

7, 8



3

8

9 v

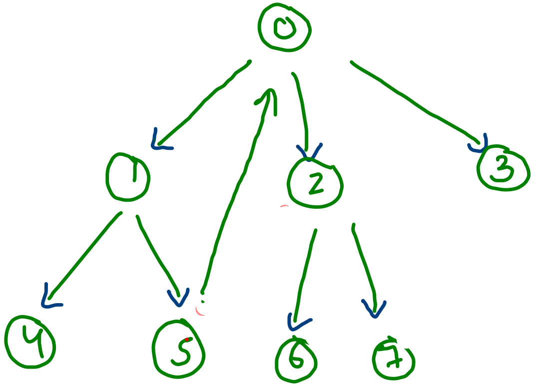
8 edges

redundant connection

a connected acyclic graph is tree.

685. Redundant Connection II

cycle



0 3

1 4

2 6

1 5 ✓

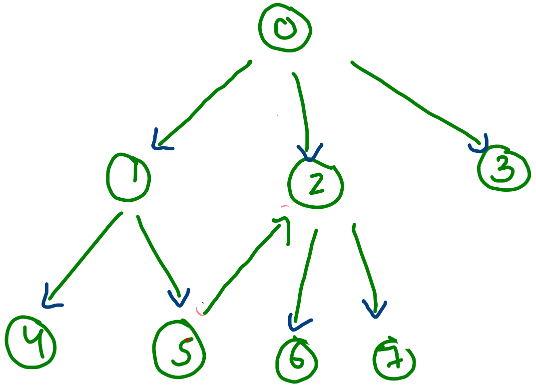
5 0 ✓

2 7

0 2

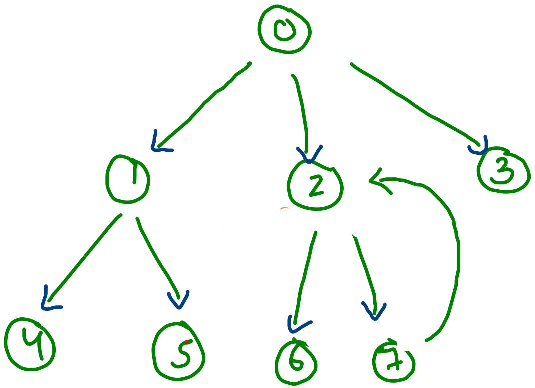
0 1 ✓

2 parent



(i) detect vtx which has 2 parent.

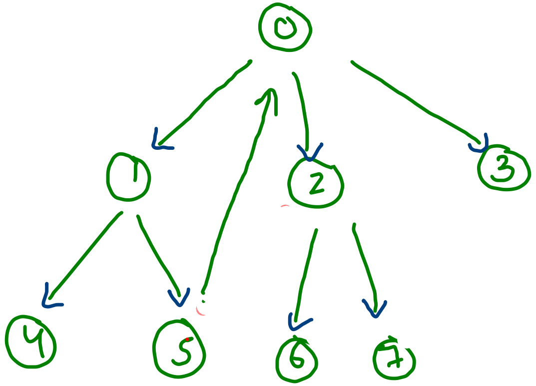
2 parent, cycle

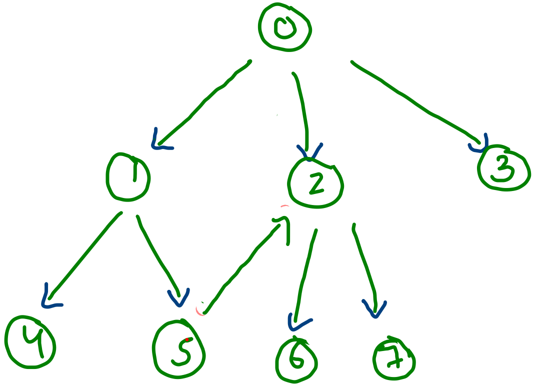


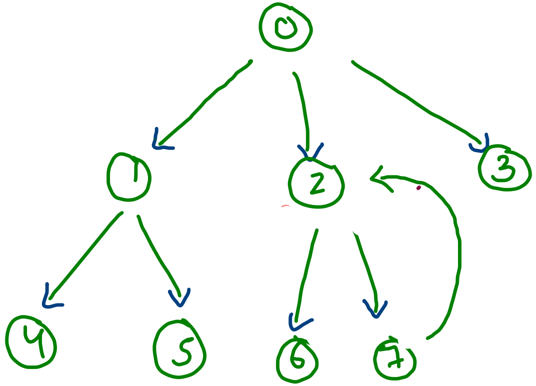
× 0 2

× 2 7

✓ 7 2







bd1

bd2

cycle

2 parent

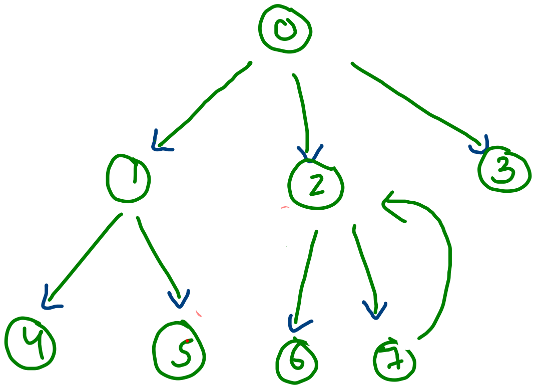
2 parent, cycle

DSU (sure : cycle, 1 cycle)

RC 1

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

2 Parent , cycle



0 -> 7 2

1 -> 0 3

2 -> 2 6

3 -> 1 4

4 -> 2 7

5 -> 0 2

6 -> 0 1

7 -> 1 5

bl1

bl2

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

```
int bl1 = -1;
int bl2 = -1;
for(int i=0; i < edges.length;i++) {
    int u = edges[i][0]; u--;
    int v = edges[i][1]; v--;

    if(indegree[v] == -1) {
        indegree[v] = i;
    }
    else {
        bl1 = indegree[v];
        bl2 = i;
        break;
    }
}
```

```
if(bl1 == -1) {
    //case 1 : cycle
    int ei = dsu(-1,edges);
    return edges[ei];
}
else {
    //case 2 : 2 parent, case3 : 2 parent & cycle
    int ei = dsu(bl2,edges);
    if(ei == -1) {
        return edges[bl2];
    }
    else {
        return edges[bl1];
    }
}
```

Eulerian path & circuit

Eulerian Path is a path in graph that visits every edge exactly once. (travel all edges exactly once)

Undirected graph : (i) if all vertices have even degree → eulerian circuit

(ii) if (V−2) vertices have even degree but any two have odd degree. → eulerian path

↳ use one of them as source and the other one as dest.

directed graph : (i) each vertex indegree == outdegree — eulerian circuit

(ii) (v-2) vertices have indegree == outdegree

two vertices, one of them should have extra indegree (dest) and the other should have extra outdegree (src).