Deadline: 29th Nov 2022

Find if Path Exists in Graph

There is a bi-directional graph with `n` vertices, where each vertex is labeled from `0` to `n - 1` (inclusive). The edges in the graph are represented as 2D integer array `edges`, where each `edge[i] = [ui, vi]` denotes a bi-directional edge between vertex `ui` and vertex `vi`. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself.

You want to determine if there is a valid path that exists from the vertex `source` to the vertex `destination`.

Given `edges` and the integers `n`, `source`, and `destination`, return `true` *if there is a valid path from* `source` *to* `destination`*, or* `false` *otherwise*.

Example 1:

```
Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2
Output: true
Explanation: There are two paths from vertex 0 to vertex 2:
- 0 → 1 → 2
- 0 → 2
```

Invert Binary Tree

Given the `root` of a binary tree, invert the tree, and return *its root*.

Example 1:

```
Input: root = [4,2,7,1,3,6,9]
Output: [4,7,2,9,6,3,1]
```