

# LeetCode 第 133 号问题：克隆图

本文首发于公众号「图解面试算法」，是 [图解 LeetCode](#) 系列文章之一。

同步博客：<https://www.algomooc.com>

题目来源于 LeetCode 上第 133 号问题：克隆图。题目难度为 Medium，目前通过率为 54.8%。

## 题目描述

给你无向连通图中一个节点的引用，请你返回该图的深拷贝（克隆）。图中的每个节点都包含它的值 val（int）和它的邻居的列表（list[Node]）。

### 示例 1:

输入: adjList = [[2,4],[1,3],[2,4],[1,3]]

输出: [[2,4],[1,3],[2,4],[1,3]]

解释:

图中有 4 个节点。

节点 1 的值是 1，它有两个邻居：节点 2 和 4。

节点 2 的值是 2，它有两个邻居：节点 1 和 3。

节点 3 的值是 3，它有两个邻居：节点 2 和 4。

节点 4 的值是 4，它有两个邻居：节点 1 和 3。

### 示例 2:

输入: adjList = [[]]

输出: [[]]

解释: 输入包含一个空列表。该图仅仅只有一个值为 1 的节点，它没有任何邻居。

### 示例 3:

输入: adjList = [[2],[1]]

输出: [[2],[1]]

## 题目解析

给你一个图，让你完整地拷贝一份。这道题目不难，但是在实际的工作项目中却时常遇到。这道题目有很多种解法，但是建议站在实际工作的角度去思考。

图是由一个个节点组成的，完整地拷贝一份图，那么就意味着我们要对每个节点进行拷贝，而且节点与节点的关系也要拷贝过来。做到这一点也不难，我们只需要遍历一遍图就可以了，这里的关键点在于题目强调是 **无向图**，也就是说我们可以从图上的任意点出发到达图上的所有节点。那么问题从而就转换到了如何遍历图，我们可以使用广度优先搜索，也可以使用深度优先搜索，从工作的角度出发，比较推荐广度优先搜索，因为理解容易，实现简单，而且不涉及栈溢出的问题，处理大规模数据比较安全。

## 动画演示

## 复杂度分析

一般使用广度优先搜索遍历图，时间复杂度是  $O(n + m)$ ，其中这里的  $n$  表示的是图上的节点数， $m$  表示的图上的边的数量。从广度优先搜索的 **由点及面** 的性质，你不难理解这个结果。极端情况下，当这张图是一张全联通的图，时间复杂度就会是  $O(n^2)$ ，解释起来也很容易，因为你每访问完一个节点，下面都会去访问相邻的节点，一个节点和所有的节点相连，那么在一个节点上花费的时间就是  $n$ ，在  $n$  节点上花费的时间就是  $n^2$ 。因为我们使用了队列存放接下来需要遍历的节点，空间复杂度就是  $O(n)$ 。