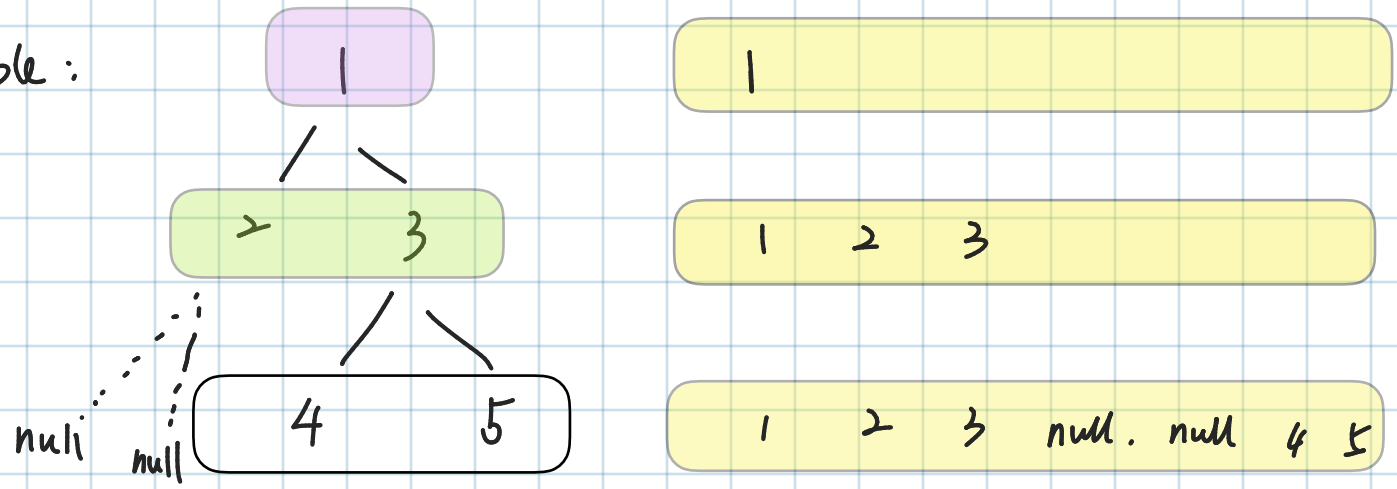
Leetcode 297: Serialize and deserialize binary trees

这道题要求将二叉树编码为一个字符串.

编码的结构和方式自选.

我个人倾向于一层一层地将二叉树编码.

example:



可使用队列:queue来完成这一层序遍历.

Serialize 代码及思路如下:

```
def serialize(self, root):
① if not root: return "[]"
    queue, res = [root], []
    while queue:
②       current = queue.pop(0)
③       if not current: res.append["null"]
        else:
            res.append(current.val)
            queue.append(current.left)
            queue.append(current.right)
④   res_str = ",".join(res)
⑤   return '['+res_str+']'
```

PS: 1: 判断边界条件：root为空结点

2: 将list对象当作queue用：pop(0)将从左起第一个元素取出元素

3: 如果当前结点为空，则用"null"标识空结点，标识是可以自选的，不一定用"null"。

4：使用逗号作为间隔符，连接 list 各元素成为一个 string

5. 返回时补上 `[` 和 `]`. 注意，这个不是必须.

deserialize 代码及思路如下：

```
def deserialize(self, data):
①   if data[1:-1] == 'null': return None
    a_list = data[1:-1].split(',')
    if not a_list: return None
②   if a_list[0] != 'null' and a_list[0] != 'None' and
       a_list[0] != '':
          root = TreeNode(int(a_list[0]))
    else: return None
    queue, i = [root], 0
③   while queue and i < len(a_list)-1:
        current = queue.pop(0)
        if current != None:
④           if a_list[i+1] != 'null':
                current.left = TreeNode(int(a_list[i+1]))
                queue.append(current.left)
            else: current.left = None
⑤           if i+2 < len(a_list):
                if a_list[i+2] != 'null':
                    current.right = TreeNode(int(a_list[i+2]))
                    queue.append(current.right)
                else: current.right = None
⑥       i = i+2
    return root
```

仍是BFS，用队列

ps:

1: "剥云" string 前后加 '[', 和 ']'

2: 这一步没必要写这么麻烦，仁者见仁

3: 为什么 i < len(arr)-1 ? 1

因为第0个元素已经是root了，我们从 i+1 个(也即0+1)元素开始遍历. 为了保证 i+1 不out of bounds, 所以要保证 i+1 < len(arr)，即 i < len(arr)-1

4: 见第5点

5: i+2 是否越界是单独判断的. 因为可能最后一个结点刚好只有一个左子结点，而没有右子结点. 第6处可以看到，i 每次递增 2.