assignment7

Assignment #7: bfs、 🌲

Updated 0851 GMT+8 Oct 21, 2025

2025 fall, Complied by 杨知进 物理学院

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

M23555: 节省存储的矩阵乘法

implementation, matrices, http://cs101.openjudge.cn/practice/23555

要求用节省内存的方式实现，不能还原矩阵的方式实现。

思路：

代码：

```
```

代码运行截图 （至少包含有"Accepted"）

M102.二叉树的层序遍历

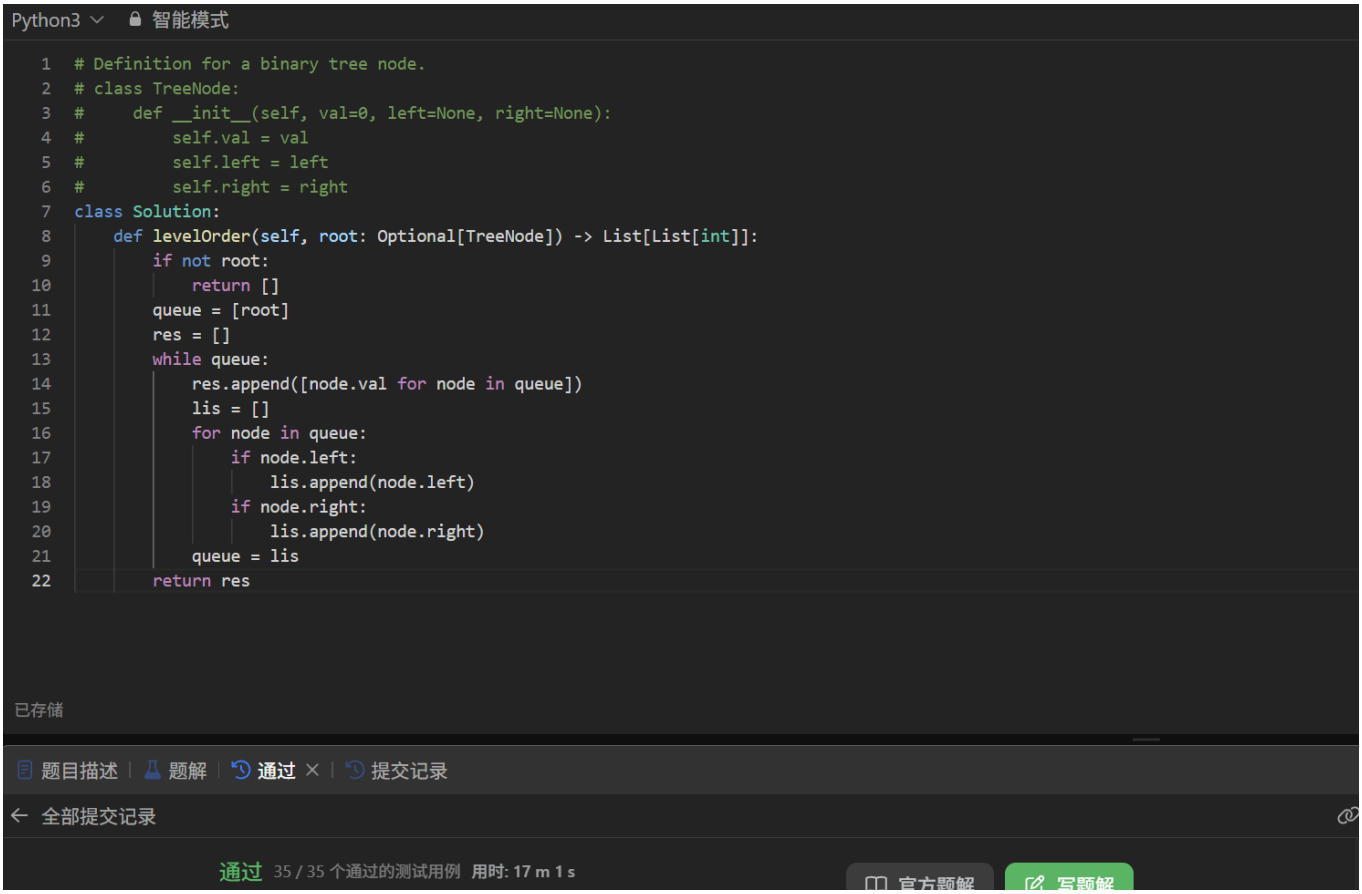
bfs, https://leetcode.cn/problems/binary-tree-level-order-traversal/

思路：

代码：

```python
class Solution:
    def levelOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
        if not root:
            return []
        queue = [root]
        res = []
        while queue:
            res.append([node.val for node in queue])
            lis = []
            for node in queue:
                if node.left:
                    lis.append(node.left)
                if node.right:
                    lis.append(node.right)
            queue = lis
        return res
```

代码运行截图 （至少包含有"Accepted"）



25fall Algo/week7/image.png

用时：14min左右

M131.分割回文串

dp, backtracking, https://leetcode.cn/problems/palindrome-partitioning/

思路：

代码：

```python
class Solution:
    def partition(self, s: str) -> List[List[str]]:
        self.check = lambda s: s == s[::-1]
        ans = []
        self.trace(s,ans,[])
        return ans
    def trace(self,s,ans,path):
        if not s:
            ans.append(path)
            return
        for i in range(1,len(s) + 1):
            if self.check(s[:i]):
                self.trace(s[i:],ans,path + [s[:i]])
```

代码运行截图 （至少包含有"Accepted"）

25fall Algo/week7/image-2.png

用时：20min左右

M200.岛屿数量

dfs, bfs, https://leetcode.cn/problems/number-of-islands/

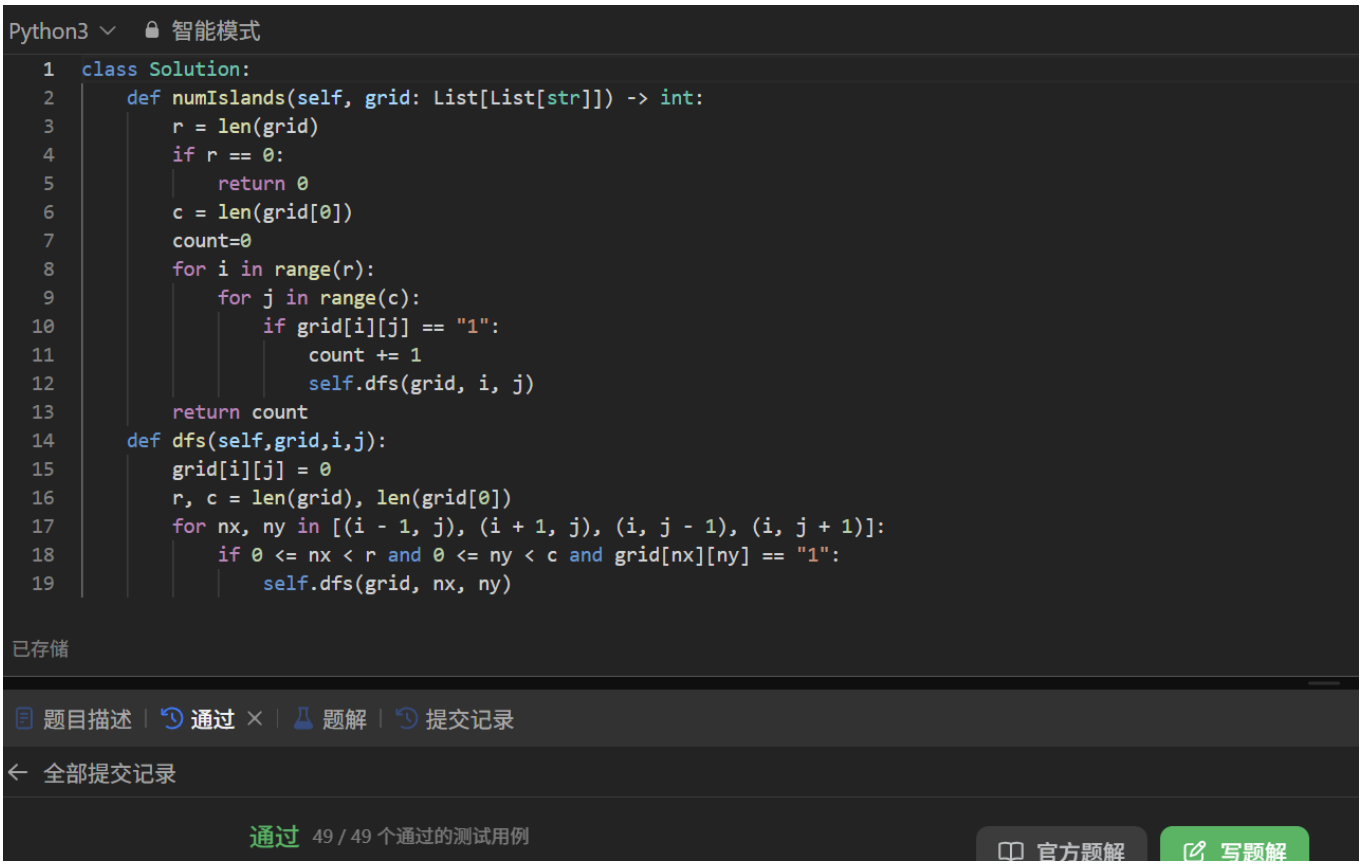
思路：

找到1就count+=1，并把这个1变成0，连在一起的也变成0；继续往后找1。

代码：

```
class Solution:
    def numIslands(self, grid: List[List[str]]) -> int:
        r = len(grid)
        if r == 0:
            return 0
        c = len(grid[0])
        count=0
        for i in range(r):
            for j in range(c):
                if grid[i][j] == "1":
                    count += 1
                    self.dfs(grid, i, j)
        return count
    def dfs(self,grid,i,j):
        grid[i][j] = 0
        r, c = len(grid), len(grid[0])
        for nx, ny in [(i - 1, j), (i + 1, j), (i, j - 1), (i, j + 1)]:
            if 0 <= nx < r and 0 <= ny < c and grid[nx][ny] == "1":
                self.dfs(grid, nx, ny)
```

（至少包含有"Accepted"）



25fall Algo/week7/image 1.png

用时：18min左右

1123.最深叶节点的最近公共祖先

dfs, https://leetcode.cn/problems/lowest-common-ancestor-of-deepest-leaves/

思路：

对每个节点来说，更深的那边是局部最优解；从下往上搜索

代码

```
class Solution:
    def lcaDeepestLeaves(self, root: Optional[TreeNode]) ->
Optional[TreeNode]:
        return self.dfs(root)[1]
    def dfs(self,node):
        if node is None:
            return 0, None
        lh, lnode = self.dfs(node.left)
        rh, rnode = self.dfs(node.right)
        if lh > rh:
            return lh + 1, lnode
        elif lh < rh:
```

```
            return rh + 1, rnode
        else:
            return lh + 1, node
```

（至少包含有"Accepted"）

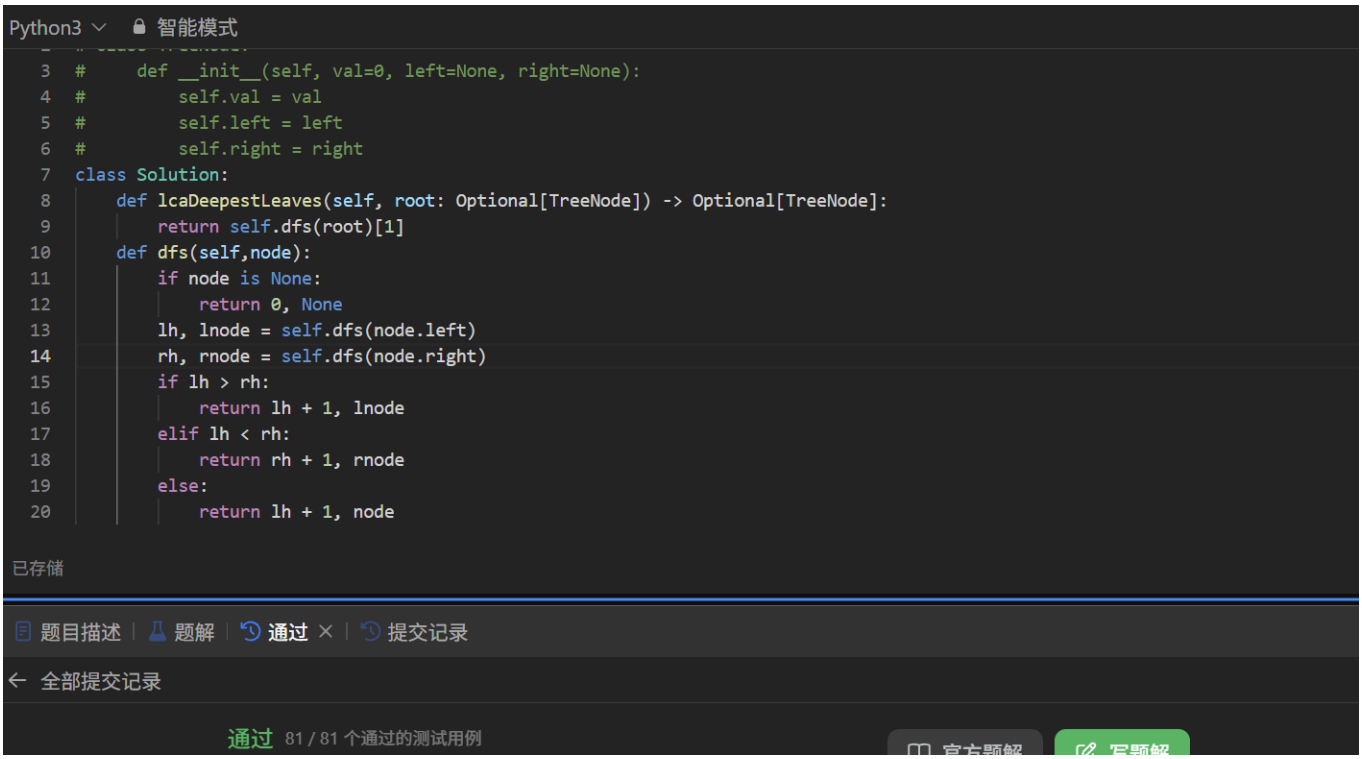


image.png

用时：30min左右

M79.单词搜索

回溯，https://leetcode.cn/problems/word-search/

思路：

代码：

```

```

代码运行截图 （至少包含有"Accepted"）

2. 学习总结和个人收获

写dfs、bfs一定程度上可以先套框架，不去纠结细节。程式化一些，然后再针对问题修改。