

# Assignment #8: 递归

Updated 1315 GMT+8 Oct 21, 2025

2025 fall, Compiled by 贺桢羽、心理与认知科学学院

## 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

### M04147汉诺塔问题(Tower of Hanoi)

dfs, <http://cs101.openjudge.cn/pctbook/M04147>

思路：递归真的好神奇。

代码

```
#M04147
def move(n, l, m, r):
    if n==1:
        print(f"1:{l}->{r}")
    else:
        move(n-1, l, r, m)
        print(f"{n}:{l}->{r}")
        move(n-1, m, l, r)
li=list(input().split())
move(int(li[0]),li[1],li[2],li[3])
```

代

#50602416提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
#M04147
def move(n, l, m, r):
    if n==1:
        print(f"1: {l}->{r}")
    else:
        move(n-1, l, r, m)
        print(f"{n}: {l}->{r}")
        move(n-1, m, l, r)
li=list(input().split())
move(int(li[0]), li[1], li[2], li[3])
```

基本信息

#: 50602416  
题目: M04147  
提交人: 25n2500013720  
内存: 3556kB  
时间: 26ms  
语言: Python3  
提交时间: 2025-10-28 16:37:54

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

码运行截图 (至少包含有"Accepted")

## M05585: 晶矿的个数

matrices, dfs similar, <http://cs101.openjudge.cn/pctbook/M05585>

思路: 学习了dfs。

代码

```
#M05585
k=int(input())
def dfs(x,y,c):
    ma[x][y]="#"
    di=[[0,1],[0,-1],[-1,0],[1,0]]
    for i in range(4):
        tx=x+di[i][0]
        ty=y+di[i][1]
        if ma[tx][ty]==c:
            dfs(tx,ty,c)
for _ in range(k):
    n=int(input())
    ma=[[0 for _ in range(n+2)]for _ in range(n+2)]
    for i in range(1,n+1):
        ma[i][1:-1]=list(input())
    r=0
    b=0
    for x in range(1,n+1):
        for y in range(1,n+1):
            if ma[x][y]=="#":
                continue
            if ma[x][y]=="r":
                dfs(x,y,"r")
                r+=1
            if ma[x][y]=="b":
                dfs(x,y,"b")
                b+=1
```

```
print(r,b)
```

代码运行截图 (至少包含有"Accepted")

#50605916提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
#M05585
k=int(input())
def dfs(x,y,c):
    ma[x][y]="#"
    di=[[0,1],[0,-1],[-1,0],[1,0]]
    for i in range(4):
        tx=x+di[i][0]
        ty=y+di[i][1]
        if ma[tx][ty]==c:
            dfs(tx,ty,c)
    for _ in range(k):
        n=int(input())
        ma=[[0 for _ in range(n+2)] for _ in range(n+2)]
        for i in range(1,n+1):
            ma[i][1:-1]=list(input())
        r=0
        b=0
        for x in range(1,n+1):
            for y in range(1,n+1):
                if ma[x][y]!="#":
                    continue
                if ma[x][y]=="r":
                    dfs(x,y,"r")
                    r+=1
                if ma[x][y]=="b":
                    dfs(x,y,"b")
                    b+=1
        print(r,b)
```

基本信息

#: 50605916  
题目: M05585  
提交人: 25n2500013720  
内存: 3696kB  
时间: 24ms  
语言: Python3  
提交时间: 2025-10-28 19:12:57

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## M02786: Pell数列

dfs, dp, <http://cs101.openjudge.cn/pctbook/M02786/>

思路: 数据较大, 要使用余数进行递推。

代码

```
#M02786
n=int(input())
dp=[0 for _ in range(1000000)]
dp[0]=1
dp[1]=2
for i in range(2,1000000):
    dp[i]=(2*dp[i-1]+dp[i-2])%32767
for i in range(n):
    print(dp[int(input())-1])
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
#M02786
n=int(input())
dp=[0 for _ in range(1000000)]
dp[0]=1
dp[1]=2
for i in range(2,1000000):
    dp[i]=(2*dp[i-1]+dp[i-2])%32767
for i in range(n):
    print(dp[int(input())-1])
```

基本信息

#: 50606130  
题目: M02786  
提交人: 25n2500013720  
内存: 40488kB  
时间: 329ms  
语言: Python3  
提交时间: 2025-10-28 19:23:38

## M46.全排列

backtracking, <https://leetcode.cn/problems/permutations/>

思路: 看了很久的视频, 终于自己能够理解回溯的意思了。

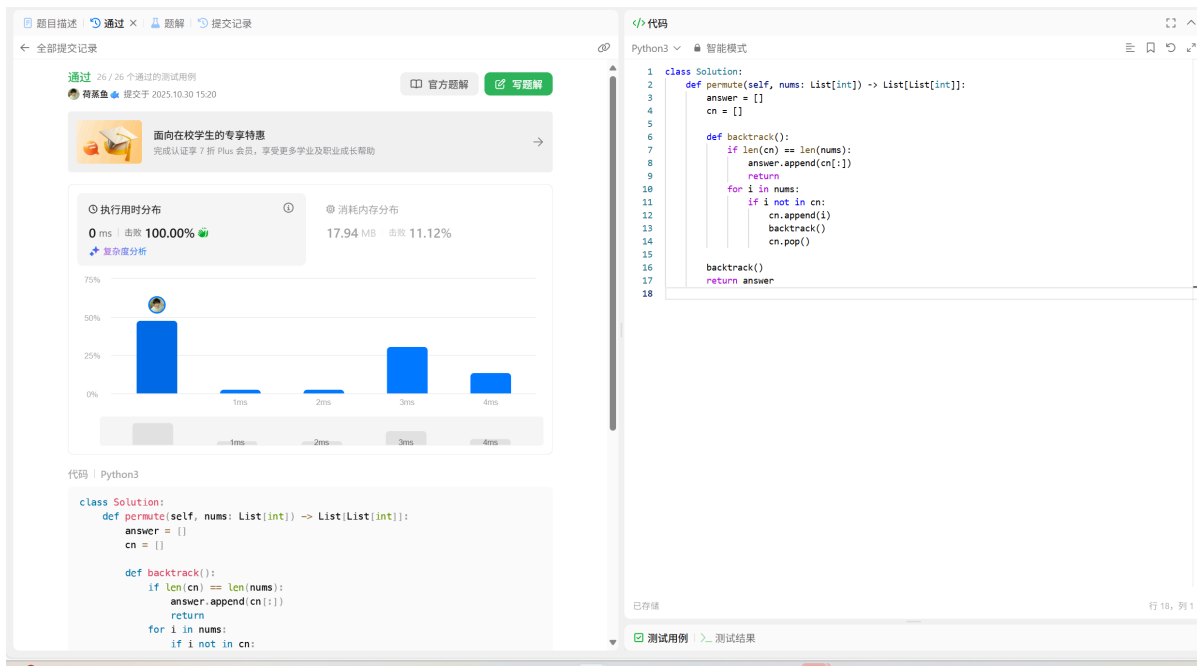
代码

```
class Solution:
    def permute(self, nums: List[int]) -> List[List[int]]:
        answer = []
        cn = []

        def backtrack():
            if len(cn) == len(nums):
                answer.append(cn[:])
                return
            for i in nums:
                if i not in cn:
                    cn.append(i)
                    backtrack()
                    cn.pop()

        backtrack()
        return answer
```

代码运行截图 (至少包含有"Accepted")



# T02754: 八皇后

dfs and similar, <http://cs101.openjudge.cn/pctbook/T02754>

思路:

代码

代码运行截图 (至少包含有"Accepted")

# T01958 Strange Towers of Hanoi

<http://cs101.openjudge.cn/practice/01958/>

思路:

代码

代码运行截图 (至少包含有"Accepted")

## 2. 学习总结和收获

---

如果作业题目简单，有否额外练习题目，比如：OJ“计概2025fall每日选做”、CF、LeetCode、洛谷等网站题目。

最近在准备高数和线代期中，加上感觉作业太难了，完全没有办法进行消化，遂只完成了四道。做这四道也花费了很多时间，我一直在试图理解递归和回溯。看了很多视频，问了很多AI，也看了题解，感觉自己对递归的掌握也越来越好了。后面剩下的两道题是一定会做的，等考完期中又要开始认真学习计概了。