

Assignment #8: 递归

Updated 1315 GMT+8 Oct 21, 2025

2025 fall, Compiled by 同学的姓名、院系

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:

代码

```
def hanoiAtoC(i, A, B, C):
    if i == 1:
        print(str(i)+":"+A+"->" + C)
    else:
        hanoiAtoB(i - 1, A, B, C)
        print(str(i) + ":" + A + "->" + C)
        hanoiBtoC(i - 1, A, B, C)

def hanoiAtoB(i, A, B, C):
    if i == 1:
```

```

        print(str(i) + ":" + A + "->" + B)
    else:
        hanoiAtoC(i - 1, A, B, C)
        print(str(i) + ":" + A + "->" + B)
        hanoiCtoB(i - 1, A, B, C)

def hanoiBtoC(i, A, B, C):
    if i == 1:
        print(str(i) + ":" + B + "->" + C)
    else:
        hanoiBtoA(i - 1, A, B, C)
        print(str(i) + ":" + B + "->" + C)
        hanoiAtoC(i - 1, A, B, C)

def hanoiCtoB(i, A, B, C):
    if i == 1:
        print(str(i) + ":" + C + "->" + B)
    else:
        hanoiCtoA(i - 1, A, B, C)
        print(str(i) + ":" + C + "->" + B)
        hanoiAtoB(i - 1, A, B, C)

def hanoiBtoA(i, A, B, C):
    if i == 1:
        print(str(i) + ":" + B + "->" + A)
    else:
        hanoiBtoC(i - 1, A, B, C)
        print(str(i) + ":" + B + "->" + A)
        hanoiCtoA(i - 1, A, B, C)

def hanoiCtoA(i, A, B, C):
    if i == 1:
        print(str(i) + ":" + C + "->" + A)
    else:
        hanoiCtoB(i - 1, A, B, C)
        print(str(i) + ":" + C + "->" + A)
        hanoiBtoA(i - 1, A, B, C)

n, a, b, c = input().split()
n = int(n)
if n <= 0:
    pass
else:
    hanoiAtoC(n, a, b, c)

```

代码运行截图 (至少包含有"Accepted")

#50684509提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
def hanoiAtoC(i, A, B, C):
    if i == 1:
        print(str(i)+":"+A+"->" + C)
    else:
        hanoiAtoB(i - 1, A, B, C)
        print(str(i) + ":" + A + "->" + C)
        hanoiBtoC(i - 1, A, B, C)

def hanoiAtoB(i, A, B, C):
    if i == 1:
```

基本信息

#: 50684509

题目: M04147

提交人: miko

内存: 3672kB

时间: 19ms

语言: Python3

提交时间: 2025-11-03 17:52:23

M05585: 晶矿的个数

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

思路:

代码

```
def f(a, s):
    if (a[0]-1, a[1]) in s:
        s.remove((a[0]-1, a[1]))
        f((a[0]-1, a[1]), s)
    if (a[0]+1, a[1]) in s:
        s.remove((a[0]+1, a[1]))
        f((a[0]+1, a[1]), s)
    if (a[0], a[1]-1) in s:
        s.remove((a[0], a[1]-1))
        f((a[0], a[1]-1), s)
    if (a[0], a[1]+1) in s:
        s.remove((a[0], a[1]+1))
        f((a[0], a[1]+1), s)

k = int(input())
for i in range(k):
    n = int(input())
    red = set()
    black = set()
    for j in range(n):
        m = input()
        for a in range(len(m)):
            if m[a] == 'r':
                red.add((j, a))
            elif m[a] == 'b':
```

```

        black.add((j, a))

    cr = 0
    cb = 0
    while True:
        try:
            r = red.pop()
            cr += 1
            f(r, red)
        except:
            break
    while True:
        try:
            b = black.pop()
            cb += 1
            f(b, black)
        except:
            break
    print(cr, cb)

```

代码运行截图 (至少包含有"Accepted")

#50620558提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def f(a, s):
    if (a[0]-1, a[1]) in s:
        s.remove((a[0]-1, a[1]))
        f((a[0]-1, a[1]), s)
    if (a[0]+1, a[1]) in s:
        s.remove((a[0]+1, a[1]))
        f((a[0]+1, a[1]), s)
    if (a[0], a[1]-1) in s:
        s.remove((a[0], a[1]-1))

```

基本信息

#: 50620558
 题目: M05585
 提交人: miko
 内存: 3720kB
 时间: 24ms
 语言: Python3
 提交时间: 2025-10-29 17:37:56

M02786: Pell数列

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

思路:

代码

```

mod = 32767
M = [[2, 1], [1, 0]]

def multiply_matrix(a, b):
    return [
        [(a[0][0]*b[0][0] + a[0][1]*b[1][0]) % mod,
         (a[0][0]*b[0][1] + a[0][1]*b[1][1]) % mod],

```

```

        [(a[1][0]*b[0][0] + a[1][1]*b[1][0]) % mod,
         (a[1][0]*b[0][1] + a[1][1]*b[1][1]) % mod]
    ]

def fast_matrix_pow(k):
    if k == 1:
        return M
    elif k%2 == 0:
        m = fast_matrix_pow(int(k/2))
        return multiply_matrix(m, m)
    else:
        return multiply_matrix(fast_matrix_pow(k-1), M)

def pell_matrix(k):
    if k == 1:
        return 1
    if k == 2:
        return 2
    m = fast_matrix_pow(k-2)
    return (m[0][0] * 2 + m[0][1] * 1) % mod

n = int(input())
for i in range(n):
    print(pell_matrix(int(input())))

```

代码运行截图（至少包含有"Accepted"）

#50623768提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

mod = 32767
M = [[2,1],[1,0]]

def multiply_matrix (a, b):
    return [
        [(a[0][0]*b[0][0] + a[0][1]*b[1][0]) % mod,
         (a[0][0]*b[0][1] + a[0][1]*b[1][1]) % mod],
        [(a[1][0]*b[0][0] + a[1][1]*b[1][0]) % mod,
         (a[1][0]*b[0][1] + a[1][1]*b[1][1]) % mod]
    ]

```

基本信息

#: 50623768
 题目: M02786
 提交人: miko
 内存: 3620kB
 时间: 24ms
 语言: Python3
 提交时间: 2025-10-29 20:12:41

M46.全排列

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

思路:

代码

```

class Solution:
    def func(a, b, c):
        if len(a) == len(c):
            b.append(a)
        else:
            for i in c:
                if i not in a:
                    d = a.copy()
                    d.append(i)
                    Solution.func(d, b, c)

    def permute(self, nums: List[int]) -> List[List[int]]:
        answer = []
        for i in nums:
            Solution.func([i], answer, nums)
        return answer

```

代码运行截图 (至少包含有"Accepted")

通过 26 / 26 个通过的测试用例

miko 提交于 2025.10.30 19:38

官方题解

写题解

🕒 执行用时分布

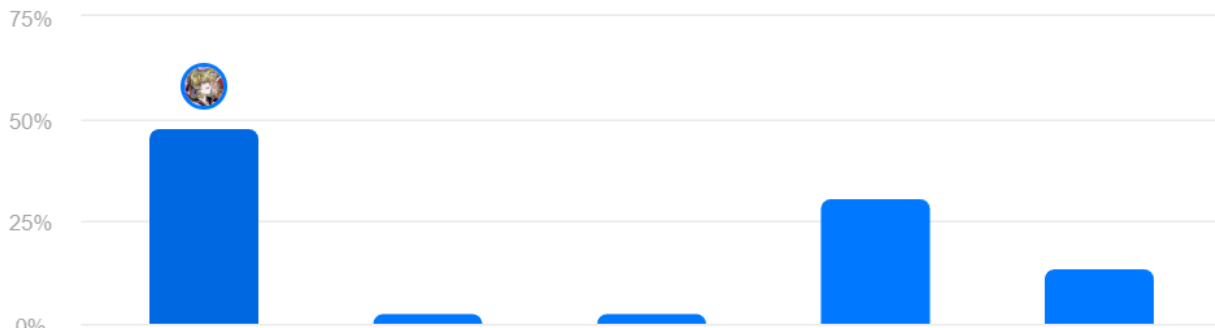


0 ms | 击败 100.00% 🏆

📊 复杂度分析

💾 消耗内存分布

17.59 MB | 击败 86.00% 🏆



T02754: 八皇后

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

思路:

代码

```

def func(a, c):
    if a // 10000000 > 0:

```

```

        c.append(a)
    else:
        b = str(a)
        for i in range(1,9):
            able = True
            for j in range(len(b)):
                if i == int(b[j]) or abs(int(b[j]) - i) == len(b) - j:
                    able = False
                    break
            if able:
                func(a*10+i, queen)

queen = []
for i in range(1,9):
    func(i, queen)
n = int(input())
for i in range(n):
    print(queen[int(input())-1])

```

代码运行截图 (至少包含有"Accepted")

#50633458提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def func(a, c):
    if a // 10000000 > 0:
        c.append(a)
    else:
        b = str(a)
        for i in range(1,9):
            able = True
            for j in range(len(b)):

```

基本信息

#: 50633458

题目: T02754

提交人: miko

内存: 3572kB

时间: 42ms

语言: Python3

提交时间: 2025-10-30 19:07:33

T01958 Strange Towers of Hanoi

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

思路:

代码

```

dp = [1]
for i in range(2,13):
    for k in range(1,i):
        try:
            dp[i-1] = min(dp[i-k-1]*2+2**k-1, dp[i-1])
        except:
            dp.append(dp[i-k-1]*2+2**k-1)

```

```
for i in dp:
    print(i)
```

代码运行截图 (至少包含有"Accepted")

#50625154提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
dp = [1]
for i in range(2,13):
    for k in range(1,i):
        try:
            dp[i-1] = min(dp[i-k-1]*2+2**k-1, dp[i-1])
        except:
            dp.append(dp[i-k-1]*2+2**k-1)
for i in dp:
    print(i)
```

基本信息

#: 50625154
题目: 01958
提交人: miko
内存: 3576kB
时间: 24ms
语言: Python3
提交时间: 2025-10-29 21:17:53

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。

无额外练习

- 1.按提示做即可，在写的时候注意逻辑，有时候自动补齐的不是你想的代码
- 2.用了set，因为递归会用大量的in
- 3.此题用了ai，我大概了解了三种方法，一种是简单递归，这种方法很慢，且容易爆栈，一种是用数组预处理，这种方法较快，但内存占用较大，第三种是用矩阵乘法，我看了ai的讲解，但ai写的递归看不懂，所以自己写了主体部分，采用了比较好懂的递归。除此之外我还用数学方法解出了数列通项，但对解题作用不大，用ai写了一个很长但用时6599ms的代码，比第一种快，但比二，三种慢
- 4.递归，难度不大
- 5.递归，难度不大
- 6.按提示做即可，实际是动态规划题