

Assignment #A : 递归、田忌赛马

Updated 2355 GMT+8 Nov 4, 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

M018160: 最大连通域面积

dfs similar, <http://cs101.openjudge.cn/pctbook/M18160>

思路：

代码

```
def f(x, y, m, c):
    m[x][y] = '.'
    c[-1] += 1
    try:
        if m[x-1][y] == 'W' and x > 0:
            f(x-1, y, m, c)
    except IndexError:
        pass
    try:
        if m[x+1][y] == 'W':
```

```

        f(x+1, y, m, c)
    except IndexError:
        pass
    try:
        if m[x][y-1] == 'W' and y > 0:
            f(x, y-1, m, c)
    except IndexError:
        pass
    try:
        if m[x][y+1] == 'W':
            f(x, y+1, m, c)
    except IndexError:
        pass
    try:
        if m[x-1][y-1] == 'W' and x > 0 and y > 0:
            f(x-1, y-1, m, c)
    except IndexError:
        pass
    try:
        if m[x-1][y+1] == 'W' and x > 0:
            f(x-1, y+1, m, c)
    except IndexError:
        pass
    try:
        if m[x+1][y+1] == 'W':
            f(x+1, y+1, m, c)
    except IndexError:
        pass
    try:
        if m[x+1][y-1] == 'W' and y > 0:
            f(x+1, y-1, m, c)
    except IndexError:
        pass

```

```

T = int(input())
for i in range(T):
    N, M = map(int, input().split())
    m = []
    for j in range(N):
        m.append(list(input()))
    c = []
    for j in range(N):
        for k in range(M):
            if m[j][k] == 'W':
                c.append(0)
                f(j, k, m, c)

```

```
if len(c) == 0:
    print(0)
else:
    print(max(c))
```

代码运行截图 (至少包含有"Accepted")

#50715517提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def f(x, y, m, c):
    m[x][y] = '.'
    c[-1] += 1
    try:
        if m[x-1][y] == 'W' and x > 0:
            f(x-1, y, m, c)
    except IndexError:
```

基本信息

#: 50715517
题目: M18160
提交人: miko
内存: 3724kB
时间: 61ms
语言: Python3
提交时间: 2025-11-05 19:02:30

sy134: 全排列III 中等

<https://sunnywhy.com/sfbj/4/3/134>

思路:

代码

```
def f(appended_num, sorted_nums, temp_dict, container, n, temp_list):

    if temp_dict[appended_num] > 0:

        tl = temp_list.copy()

        tl.append(appended_num)

        td = temp_dict.copy()

        td[appended_num] -= 1

        if len(tl) == n:

            container.append(tl)

    else:

        for i in sorted_nums:

            f(i, sorted_nums, td, container, n, tl)
```

```
n = int(input())

nums = list(map(int, input().split()))

DICT = {}

for i in nums:

    if i in DICT:

        DICT[i] += 1

    else:

        DICT[i] = 1

nums = sorted(list(DICT.keys()))

c = []

for i in nums:

    f(i, nums, DICT, c, n, [])

for i in c:

    print(' '.join(str(j) for j in i))
```

代码运行截图 (至少包含有"Accepted")

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1 def f(append_num, sorted_nums, temp_dict, container):
2     if temp_dict[append_num] > 0:
3         t1 = temp_list.copy()
4         t1.append(append_num)
5         td = temp_dict.copy()
6         td[append_num] -= 1
7         if len(t1) == n:
8             container.append(t1)
```

sy136: 组合II 中等

<https://sunnywhy.com/sfbj/4/3/136>

给定一个长度为n的序列，其中有n个互不相同的正整数，再给定一个正整数k，求从序列中任选k个的所有可能结果。

思路：

代码

```
def f(n, k, nums, temp_list, c, temp_index):

    if len(temp_list) == k:

        c.append(temp_list)

    else:

        for i in range(temp_index, n-(k-len(temp_list)-1)):

            a = temp_list.copy()

            a.append(nums[i])

            f(n, k, nums, a, c, i+1)
```

```

n, k = map(int, input().split())

nums = sorted(list(map(int, input().split())))

c = []

f(n, k, nums, [], c, 0)

for i in range(len(c)):

    print(" ".join(map(str, c[i])))

```

代码运行截图 (至少包含有"Accepted")

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```

1  def f(n, k, nums, temp_list, c, temp_index):
2      if len(temp_list) == k:
3          c.append(temp_list)

```

sy137: 组合III 中等

<https://sunnywhy.com/sfbj/4/3/137>

思路:

代码

```

def f(n, k, nums, temp_list, c, temp_index):

    if len(temp_list) == k:

        if temp_list not in c:

            c.append(temp_list)

```

```

else:

    for i in range(temp_index, n-(k-len(temp_list)-1)):

        a = temp_list.copy()

        a.append(nums[i])

        f(n, k, nums, a, c, i+1)

```

```

n, k = map(int, input().split())

nums = sorted(list(map(int, input().split())))

c = []

f(n, k, nums, [], c, 0)

for i in range(len(c)):

    print(" ".join(map(str, c[i])))

```

代码运行截图 (至少包含有"Accepted")

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```

1  def f(n, k, nums, temp_list, c, temp_index):
2      if len(temp_list) == k:
3          if temp_list not in c:
4              c.append(temp_list)
5      else:
6          for i in range(temp_index, n-(k-len(temp_list)-1)):
7              a = temp_list.copy()

```

M04123: 马走日

dfs, <http://cs101.openjudge.cn/pctbook/M04123>

思路:

代码

```
def f(n, m, x, y, c, t):
    t += 1
    c[x][y] = 1
    if n*m == t:
        return 1
    else:
        answer = 0
        if 0 <= x-2 <= n-1 and 0 <= y-1 <= m-1 and c[x-2][y-1] == 0:
            answer += f(n, m, x-2, y-1, c, t)
            c[x-2][y-1] = 0
        if 0 <= x-2 <= n-1 and 0 <= y+1 <= m-1 and c[x-2][y+1] == 0:
            answer += f(n, m, x-2, y+1, c, t)
            c[x-2][y+1] = 0
        if 0 <= x+2 <= n-1 and 0 <= y-1 <= m-1 and c[x+2][y-1] == 0:
            answer += f(n, m, x+2, y-1, c, t)
            c[x+2][y-1] = 0
        if 0 <= x+2 <= n-1 and 0 <= y+1 <= m-1 and c[x+2][y+1] == 0:
            answer += f(n, m, x+2, y+1, c, t)
            c[x+2][y+1] = 0
        if 0 <= x-1 <= n-1 and 0 <= y-2 <= m-1 and c[x-1][y-2] == 0:
            answer += f(n, m, x-1, y-2, c, t)
            c[x-1][y-2] = 0
        if 0 <= x-1 <= n-1 and 0 <= y+2 <= m-1 and c[x-1][y+2] == 0:
            answer += f(n, m, x-1, y+2, c, t)
            c[x-1][y+2] = 0
        if 0 <= x+1 <= n-1 and 0 <= y-2 <= m-1 and c[x+1][y-2] == 0:
            answer += f(n, m, x+1, y-2, c, t)
            c[x+1][y-2] = 0
        if 0 <= x+1 <= n-1 and 0 <= y+2 <= m-1 and c[x+1][y+2] == 0:
            answer += f(n, m, x+1, y+2, c, t)
            c[x+1][y+2] = 0
        return answer

T = int(input())
for i in range(T):
    n, m, x, y = map(int, input().split())
    c = []
    for j in range(n):
        c.append([0]*m)
```



```
t = 0
answer = f(n, m, x, y, c, t)
print(answer)
```

代码运行截图 (至少包含有"Accepted")

#50854412提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def f(n, m, x, y, c, t):
    t += 1
    c[x][y] = 1
    if n*m == t:
        return 1
    else:
        answer = 0
        if 0 <= x-2 <= n-1 and 0 <= y-1 <= m-1 and c[x-2][y-1] == 0:
            answer += f(n, m, x-2, y-1, c, t)
        c[x-2][y-1] = 0
```

基本信息

#: 50854412
题目: M04123
提交人: miko
内存: 3784kB
时间: 2977ms
语言: Python3
提交时间: 2025-11-15 17:49:23

T02287: Tian Ji -- The Horse Racing

greedy, dfs <http://cs101.openjudge.cn/pctbook/T02287>

思路:

解题思路

- 排序处理:** 将田忌和齐王的马的速度分别按**升序**排序, 以便从“最慢”和“最快”两个维度进行策略性匹配。
- 双指针策略:** 使用四个指针分别指向田忌和齐王的“最慢马”和“最快马”, 分三种情况进行匹配:
 - 若田忌最快的马能赢齐王最快的马, 直接匹配这两匹, 赢一场。
 - 若田忌最快的马输于齐王最快的马, 用田忌最慢的马去输这场, 消耗齐王最快的马。
 - 若田忌最快的马与齐王最快的马速度相同, 再比较双方最慢的马:
 - 若田忌最慢的马能赢齐王最慢的马, 匹配这两匹, 赢一场。
 - 否则, 用田忌最慢的马与齐王最快的马匹配 (输或平局), 消耗对方最快的马。

代码

```
while True:
    n = int(input())
    if n == 0:
        break
    tian = list(map(int, input().split()))
    king = list(map(int, input().split()))
    # 升序排序
```

```

tian.sort()
king.sort()
# 初始化指针: i指向最慢, j指向最快
i_t, j_t = 0, n - 1
i_k, j_k = 0, n - 1
win = 0
lose = 0
while i_t <= j_t:
    if tian[j_t] > king[j_k]:
        # 田忌最快赢齐王最快
        win += 1
        j_t -= 1
        j_k -= 1
    elif tian[j_t] < king[j_k]:
        # 田忌最快输, 用最慢输齐王最快
        lose += 1
        i_t += 1
        j_k -= 1
    else:
        # 田忌最快与齐王最快速度相同
        if tian[i_t] > king[i_k]:
            # 田忌最慢赢齐王最慢
            win += 1
            i_t += 1
            i_k += 1
        else:
            # 田忌最慢不赢, 用最慢输齐王最快
            if tian[i_t] < king[j_k]:
                lose += 1
            i_t += 1
            j_k -= 1
# 收益 = (赢的场次 - 输的场次) * 200
profit = (win - lose) * 200
print(profit)

```

代码运行截图（至少包含有"Accepted"）

#50894996提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
while True:
    n = int(input())
    if n == 0:
        break
    tian = list(map(int, input().split()))
    king = list(map(int, input().split()))
    # 升序排序
    tian.sort()
    king.sort()
    # 初始化指针: i指向最慢, j指向最快
    i_t, j_t = 0, n - 1
    i_k, j_k = 0, n - 1
```

基本信息

#: 50894996
题目: T02287
提交人: miko
内存: 3816kB
时间: 54ms
语言: Python3
提交时间: 2025-11-18 19:23:55

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.了解到一种WA的原因：X=0的时候，访问数组【X-1】的元素不会报错，而是直接倒序访问
- 2.尝试使用字典进行优化，但貌似并没有起到优化效果，一是测试数据小，二是我当时还不会回溯，导致代码有问题（当然现在会了）
- 3.无
- 4.请问最好的地方就在于你能看到出错的样例，你的输出和正确的输出，我在3的基础上加了一行就过了
- 5.用到了回溯，感觉可以再优化
- 6.用了ai，了解了双指针和贪心