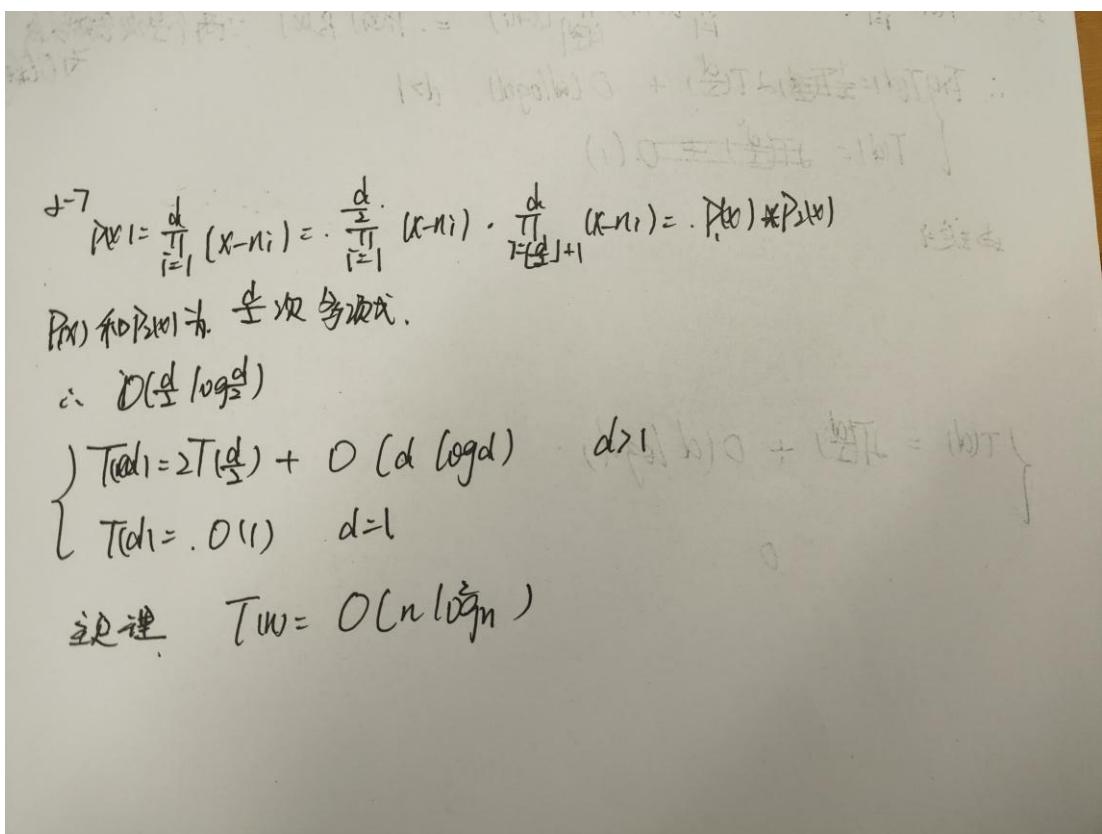


2-7



2-15

```
import numpy as np
n = int(input())
a = np.zeros((n, n))
b = np.zeros(n)

def tournament(n):
    if n == 1:
        a[1][1] = 1
        return
    if n%2 == 1:
        tournament(n+1)
        return
    tournament(n//2)
    makecopy(n)

def makecopy(n):
    if n//2 > 1 or n%2 == 1:
        copyodd(n)
    else:
        copy(n)

def copy(n):
    m = n//2
```

```

for i in range(1,m+1):
    for j in range(1,m+1):
        a[i][j+m] = a[i][j] + m
        a[i+m][j] = a[i][j+m]
        a[i+m][j+m] = a[i][j]

def copyodd(n):
    m = n//2
    for i in range(1,m+1):
        b[i] = m + i
        b[i+m] = b[i]
    for i in range(1,m+1):
        for j in range(1,m+2):
            if a[i][j]>m:
                a[i][j] = b[i]
                a[m+i][j] = (b[i] + m) % n
            else:
                a[m+i][j] = a[i][j] + m

        for j in range(2,m+1):
            a[i][m+j] = b[i+j-1]
            a[b[i+j-1]][m+j] = i

```

2-6

```

import math
# 字典序排列的 rank
def permrank (n,strpi):
    if n == 1:
        return 1
    else:
        substr = strpi[1:n+1]
        adjusted_substr = ''.join(
            str(int(char) - 1) if int(char) > int(strpi[0]) else char
            for char in substr)

        return permrank(n-1,adjusted_substr) + (int(strpi[0])-1) * math.factorial(n-1)
def permnext(stri):#找到字典序的下一个排列
    arr = list(map(int,stri))
    n = len(arr)
    i = n-2
    while i >= 0 and arr[i] >= arr[i+1]:
        i -= 1
    if i == -1:

```

```

        return arr
j = n-1
while arr[j] <= arr[i]:
    j -= 1
arr[i], arr[j] = arr[j], arr[i]
arr[i+1:] = arr[n-1:i:-1]
str2 = ''.join(map(str,arr))
return str2
filename = 'permutations.txt'
with open(filename) as f:
    #第一行是n
    n = int(f.readline())
    #第二行是排列的第一个数
    strpi = f.readline().strip()
    #打开文件输出排列的rank
    with open('permutations_rank.txt','w') as fw:
        fw.write(str(permrank(n,strpi)-1))
        fw.write('\n')
        fw.write(str(permnext(strpi)))

```

## 2-9

```

def hanoi(n, a, b, c):
    if n == 1:
        print(f" {1}{a} {c}")  # 将第n个圆盘从a移动到c
    else:
        hanoi(n-1, a, c, b)  # 将前n-1个圆盘从a移动到b, 使用c作为辅助
        print(f" {n}{a} {c}")  # 将第n个圆盘从a移动到c

        a_top.pop()
        c_top.append(n)

        hanoi(n-1, b, a, c)  # 将n-1个圆盘从b移动到c, 使用a作为辅助

```



同济大学  
TONGJI UNIVERSITY  
SHANGHAI  
PEOPLE'S REPUBLIC OF CHINA

证明：

假设 固盘个数为 n, hanoi(n, A, B, C) 不违反规则(3) 且在移动过程中 且是低固盘与 n 的转轴与 n 不同奇偶。  
当固盘个数为 n+1.

(1) hanoi(n+1, A, C, B)

(2) move (A, B)

(3) hanoi(n+1, C, B, A)

与 (1) 不违反规则(3) C 上最低固盘与 n 同奇偶 B 与 n 不同奇偶 从而  
C 与 n 不同奇偶 B 与 n 同奇偶

与 (2) 不违反且 B 上最低固盘与 n 同奇偶

与 (3) 不违反规则(3) 且移动过程中 B 上的最高 2 个固盘的奇偶性与 n  
相同奇偶性 A 上是低固盘不同奇偶性 从而 A 上的最高 2 个固盘  
奇偶性与 n 具有不同奇偶性 故在 A 上最低固盘的奇偶性与 n 具有相同  
奇偶性

由数学归纳法知，hanoi(n, A, B, C) 不违反规则 3.