# Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Complied by 陈俊逸，工学院

**说明：**

1）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3）提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

4）如果不能在截止前提交作业，请写明原因。

## 1. 题目

### sy119: 汉诺塔（20min左右）

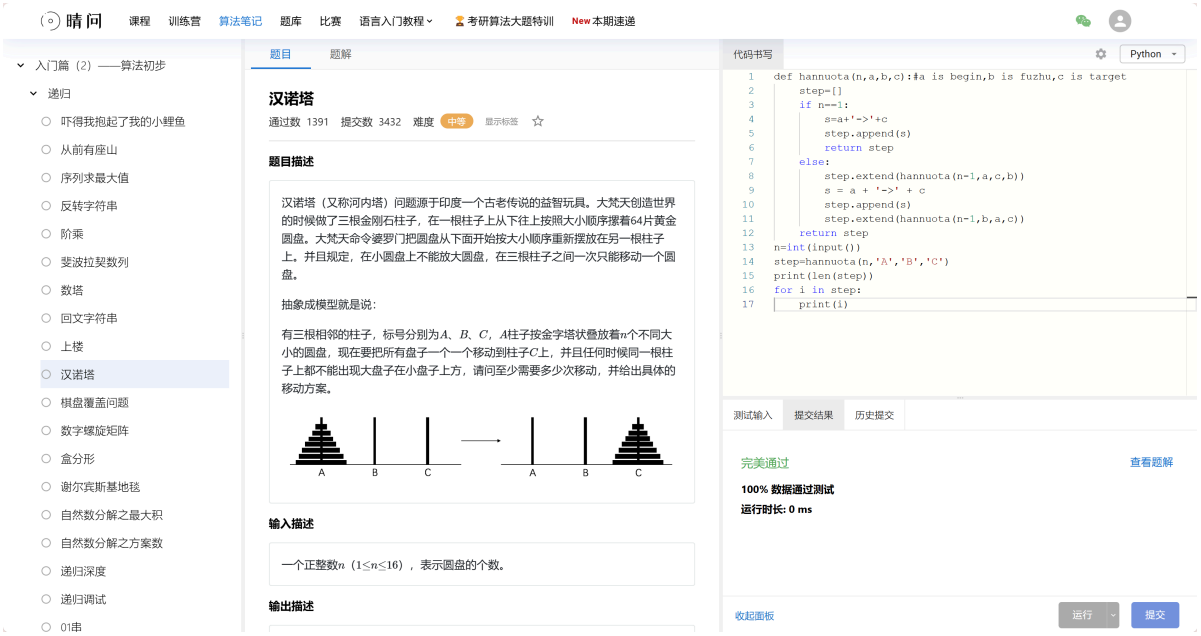
recursion, https://sunnywhy.com/sfbj/4/3/119

思路：递归，先把n-1全放到b上，把第n个放到c上，再把n-1全放到c上即可

代码：

```
def hannuota(n,a,b,c):#a is begin,b is fuzhu,c is target
    step=[]
    if n==1:
        s=a+'->'+c
        step.append(s)
        return step
    else:
        step.extend(hannuota(n-1,a,c,b))
        s = a + '->' + c
        step.append(s)
        step.extend(hannuota(n-1,b,a,c))
    return step
n=int(input())
step=hannuota(n,'A','B','C')
print(len(step))
for i in step:
    print(i)
```

代码运行截图 （至少包含有"Accepted"）




# sy132: 全排列I（1h)

recursion, https://sunnywhy.com/sfbj/4/3/132

思路:

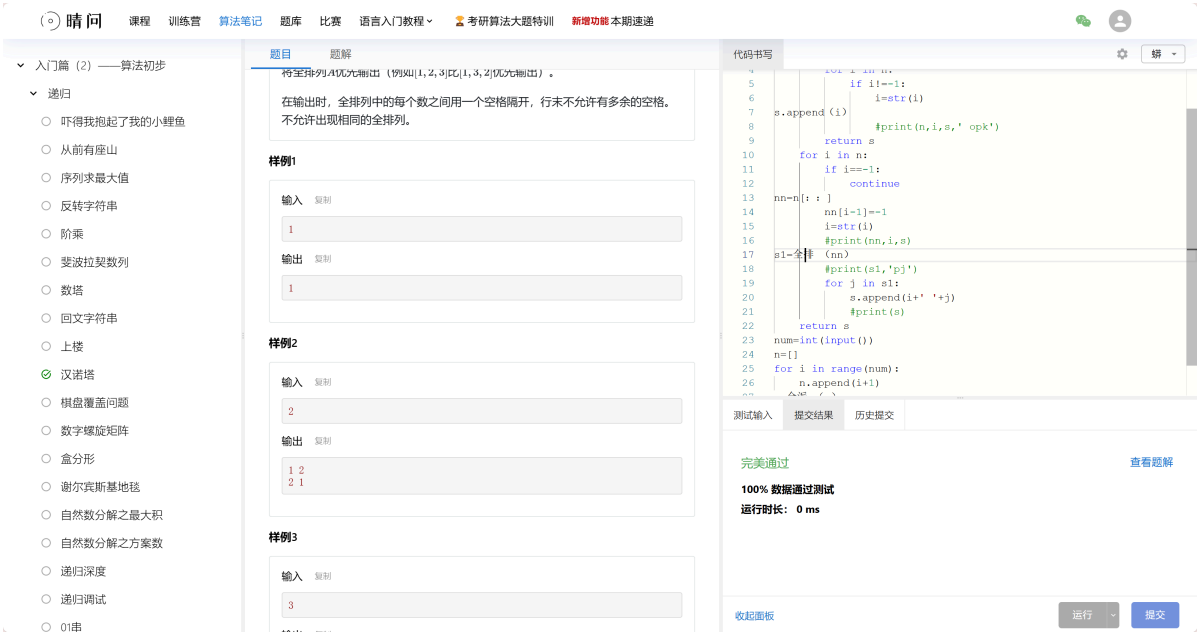
把n个数字遍历一遍，后面再加上除去这个数字的n-1递归排列

代码:

```
def quanpai(n):
    s=[]
    if len(n)-1==n.count(-1):
        for i in n:
            if i!=-1:
                i=str(i)
                s.append(i)
                #print(n,i,s,' opk')
        return s
    for i in n:
        if i==-1:
            continue
        nn=n[::]
        nn[i-1]=-1
        i=str(i)
        #print(nn,i,s)
        s1=quanpai(nn)
        #print(s1,'pj')
        for j in s1:
            s.append(i+' '+j)
            #print(s)
    return s
num=int(input())
n=[]
for i in range(num):
    n.append(i+1)
s=quanpai(n)
```

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

代码运行截图 ==（至少包含有"Accepted"）==




## 02945: 拦截导弹 （long long int time；）

dp, http://cs101.openjudge.cn/2024fallroutine/02945

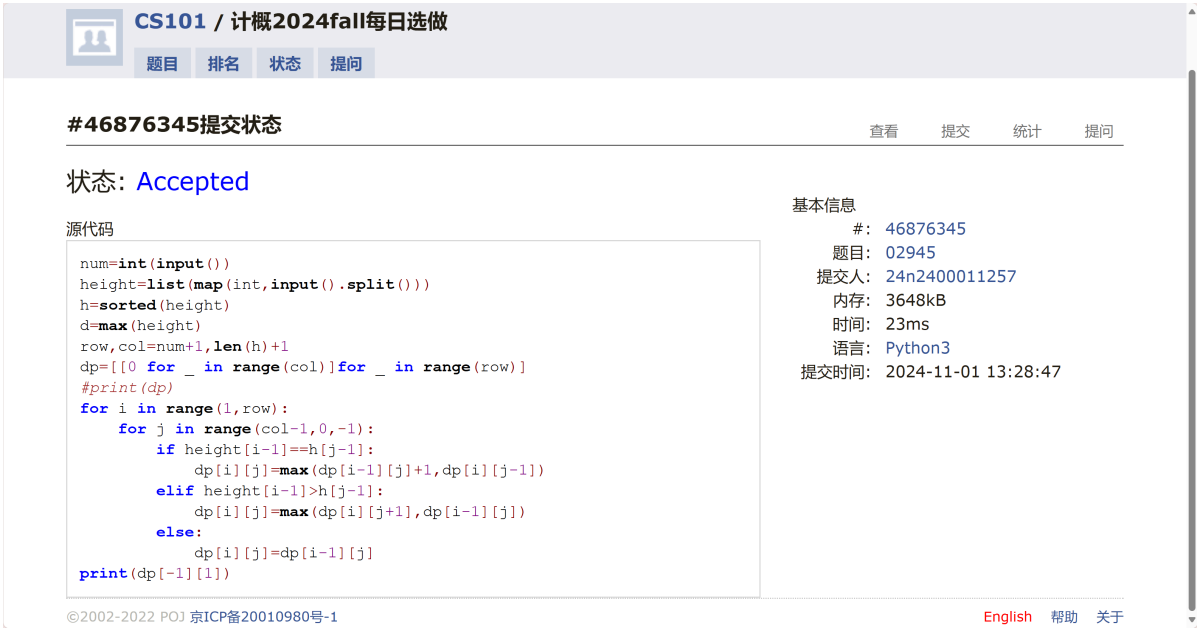
思路：创建一个二维数组，行为导弹顺序，列为高度（准确来说，要考虑的列是发射当前导弹后小于等于当前高度的一系列值），列指针反着跑，如果行列数相等，说明导弹可以选择发射与否，这个格应该考虑：上一格加1，和右一格的最大值（所以这里需要列指针反着跑）。如果行指标大，一样的道理（只是此时不能发射，故上一种情况不用加1即可）。如果列指标大，类似于一种indexerror，不管它，让它为0就行。

最后输出最后一行的第一个数即可（一定是最后一行里最大的）

代码：

```
num=int(input())
height=list(map(int,input().split()))
h=sorted(height)
d=max(height)
row,col=num+1,len(h)+1
dp=[[0 for _ in range(col)]for _ in range(row)]
#print(dp)
for i in range(1,row):
    for j in range(col-1,0,-1):
        if height[i-1]==h[j-1]:
            dp[i][j]=max(dp[i-1][j]+1,dp[i][j-1])
        elif height[i-1]>h[j-1]:
            dp[i][j]=max(dp[i][j+1],dp[i-1][j])
        else:
            dp[i][j]=dp[i-1][j]
print(dp[-1][1])
```

代码运行截图 （至少包含有"Accepted"）




## 23421: 小偷背包 （10min）

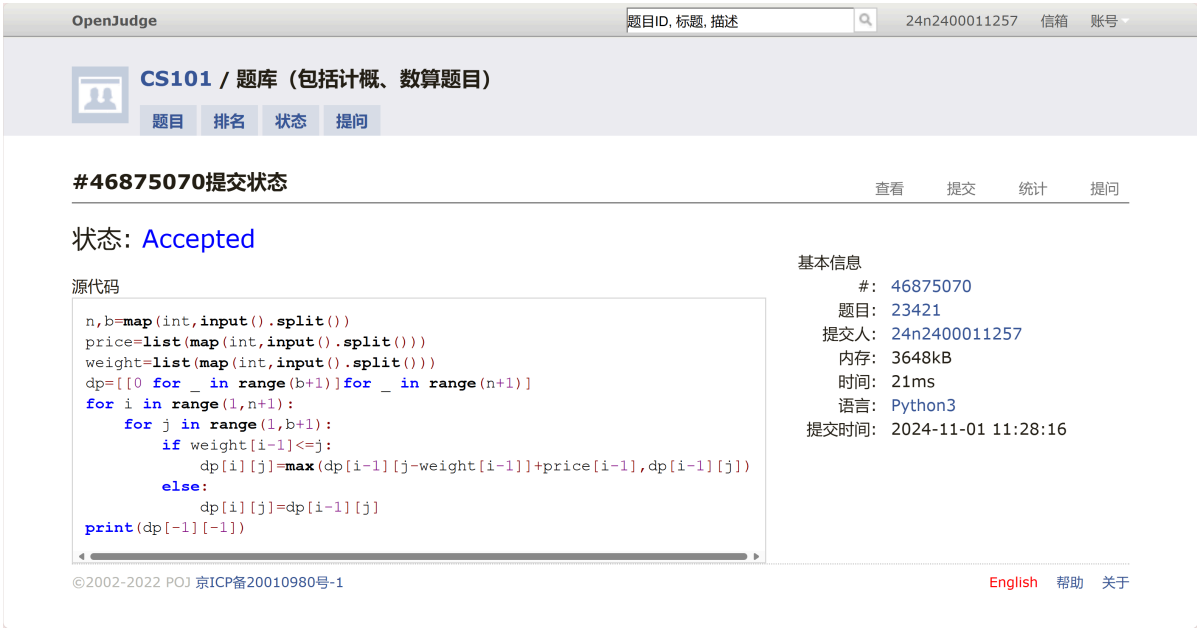
dp, http://cs101.openjudge.cn/practice/23421

思路：一件一件商品考虑，一格一格容量出发，每一格对应的最优解为，考虑拿取当前商品后剩余容量在上一行的最优解，和不拿取在上一行的最优解，二者进行取max，得到最终的最优解。（详细打一遍思路强化一下印象）

代码：

```
n,b=map(int,input().split())
price=list(map(int,input().split()))
weight=list(map(int,input().split()))
dp=[[0 for _ in range(b+1)]for _ in range(n+1)]
for i in range(1,n+1):
    for j in range(1,b+1):
        if weight[i-1]<=j:
            dp[i][j]=max(dp[i-1][j-weight[i-1]]+price[i-1],dp[i-1][j])
        else:
            dp[i][j]=dp[i-1][j]
print(dp[-1][-1])
```

代码运行截图 （至少包含有"Accepted"）




# 02754: 八皇后（两小时左右）

dfs and similar, http://cs101.openjudge.cn/practice/02754

思路：

函数t用来判断放完一个queen后哪些格子不能用了

search第一个递归，让它去找能放queen的可能位置

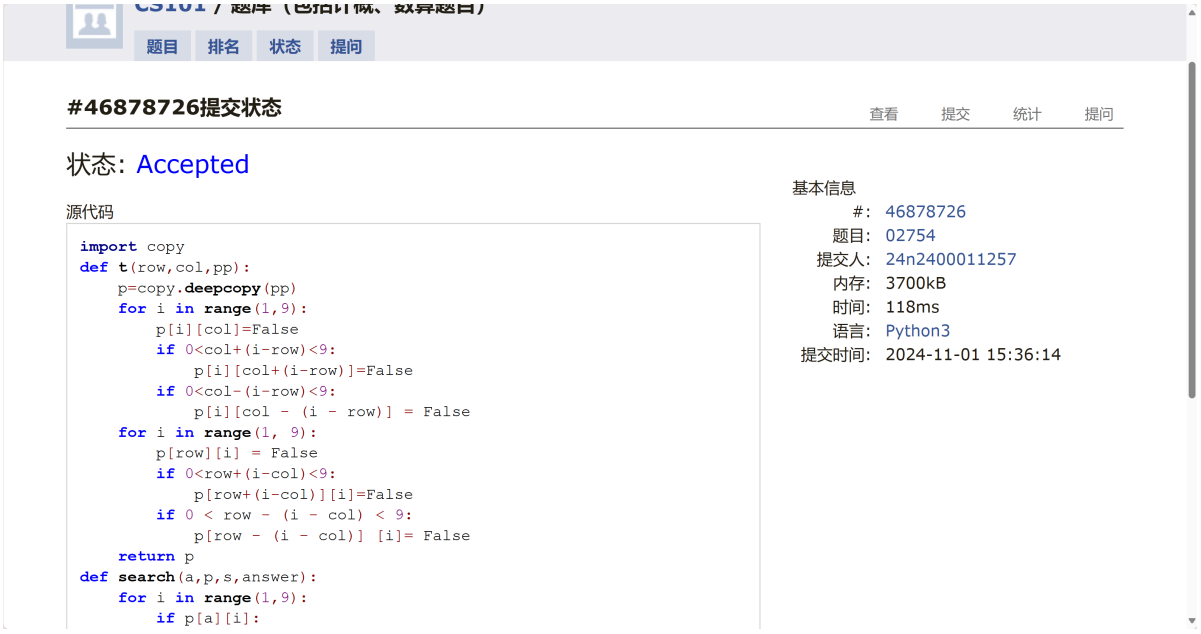
s用于记录位置，answer用于存储符合条件（长度为8）的s

代码：

```
import copy
def t(row,col,pp):
    p=copy.deepcopy(pp)
    for i in range(1,9):
        p[i][col]=False
        if 0<col+(i-row)<9:
            p[i][col+(i-row)]=False
        if 0<col-(i-row)<9:
            p[i][col - (i - row)] = False
    for i in range(1, 9):
        p[row][i] = False
        if 0<row+(i-col)<9:
            p[row+(i-col)][i]=False
        if 0 < row - (i - col) < 9:
            p[row - (i - col)] [i]= False
    return p
def search(a,p,s,answer):
    for i in range(1,9):
        if p[a][i]:
            s+=str(i)
            #print(s)
            if a==8:
                answer.append(s)
            else:
```

```
                search(a + 1,t(a, i, p), s, answer)
            s=s[:-1]
    return answer
s=''
p=[[True for i in range(9)] for j in range(9)]
answer=[]
answer=search(1,p,s,answer)
#print(answer)
n=int(input())
for i in range(n):
    shuru=int(input())
    print(answer[shuru-1])
```

代码运行截图 （至少包含有"Accepted"）




## 189A. Cut Ribbon （一个半小时）

brute force, dp 1300 https://codeforces.com/problemset/problem/189/A

思路：

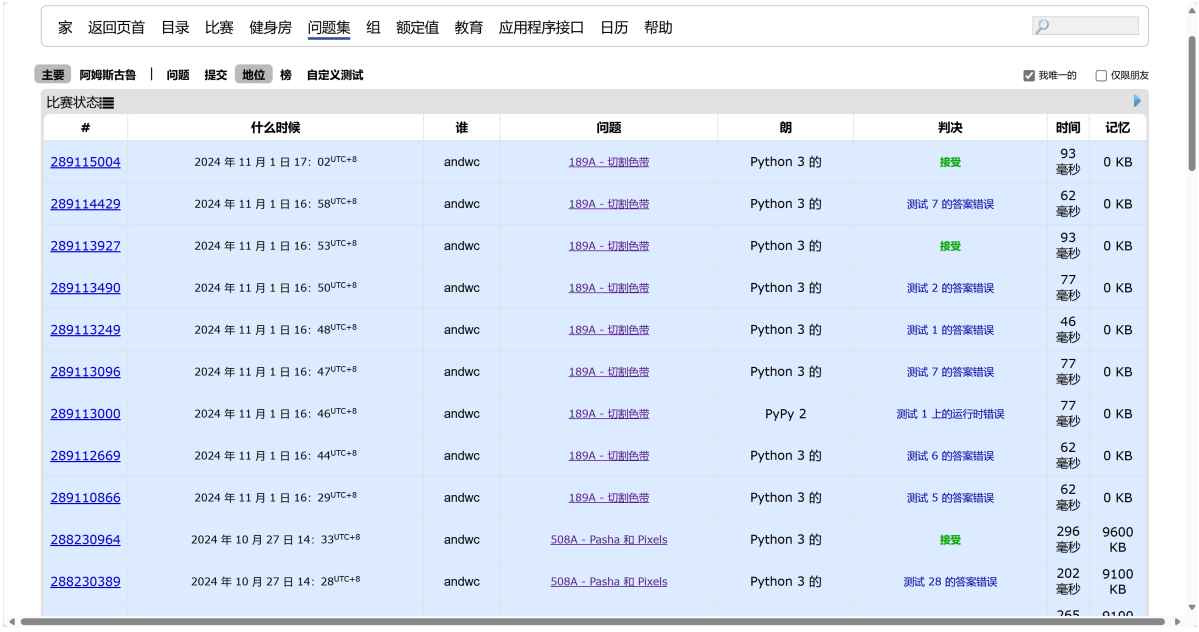
本题思路自己想的时候一直没想明白，借助ai后恍然大悟

一维数组，坐标表示现有长度，从1遍历到n，假设剪了a（或b，c），看是否符合条件（即剩下的长度是否有解），不断dp即可。

代码：

```python
shuru = list(map(int, input().split()))
n = shuru[0]
a, b, c = sorted(shuru[1:4])
dp = [-1] * (n + 1)
dp[0] = 0
for i in range(1, n + 1):
    if i >= a and dp[i - a] != -1:
        dp[i] = max(dp[i], dp[i - a] + 1)
    if i >= b and dp[i - b] != -1:
        dp[i] = max(dp[i], dp[i - b] + 1)
    if i >= c and dp[i - c] != -1:
        dp[i] = max(dp[i], dp[i - c] + 1)
print(dp[n])
```

代码运行截图 （至少包含有"Accepted"）




| # | 什么时候 | 谁 | 问题 | 朗 | 判决 | 时间 | 记忆 |
|---|---|---|---|---|---|---|---|
| 289115004 | 2024 年 11 月 1 日 17: 02UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 接受 | 93 毫秒 | 0 KB |
| 289114429 | 2024 年 11 月 1 日 16: 58UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 7 的答案错误 | 62 毫秒 | 0 KB |
| 289113927 | 2024 年 11 月 1 日 16: 53UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 接受 | 93 毫秒 | 0 KB |
| 289113490 | 2024 年 11 月 1 日 16: 50UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 2 的答案错误 | 77 毫秒 | 0 KB |
| 289113249 | 2024 年 11 月 1 日 16: 48UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 1 的答案错误 | 46 毫秒 | 0 KB |
| 289113096 | 2024 年 11 月 1 日 16: 47UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 7 的答案错误 | 77 毫秒 | 0 KB |
| 289113000 | 2024 年 11 月 1 日 16: 46UTC+8 | andwc | 189A - 切割色带 | PyPy 2 | 测试 1 上的运行时错误 | 77 毫秒 | 0 KB |
| 289112669 | 2024 年 11 月 1 日 16: 44UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 6 的答案错误 | 62 毫秒 | 0 KB |
| 289110866 | 2024 年 11 月 1 日 16: 29UTC+8 | andwc | 189A - 切割色带 | Python 3 的 | 测试 5 的答案错误 | 62 毫秒 | 0 KB |
| 288230964 | 2024 年 10 月 27 日 14: 33UTC+8 | andwc | 508A - Pasha 和 Pixels | Python 3 的 | 接受 | 296 毫秒 | 9600 KB |
| 288230389 | 2024 年 10 月 27 日 14: 28UTC+8 | andwc | 508A - Pasha 和 Pixels | Python 3 的 | 测试 28 的答案错误 | 202 毫秒 | 9100 KB |

# 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2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

初次尝试递归，尤其是自己独立想出汉诺塔的代码后，我正真体会到了递归的优雅，真是太优雅了，华丽又简洁，朴素却暗含力量，太有意思了（而且真正想懂了思路也不用过于费时间，我是第一次尝试时间较长可能是因为有点生疏）

全排列一题感觉思路不是很难想，但是代码实现起来总是会出现各种问题，最后ac了，也算是一种提升。并且在看完题解后很受启发。

八皇后，通过ai，我学会了深浅拷贝的区别，以及回溯的用法。

最后一题收获也比较大，写了一个半小时左右，做不出来，问了ai，它给出了只有一维数组的解法，很简洁，很优雅，很牛，而且是正确的（我用二维数组也没想明白，但是他这个代码就很明了），同时也让我学会了一些初始化list的小技巧。

dp的最后几题太难了（对我来说），刚刚接触完全不会，下来又花了很大的功夫，读了算法图解以及很多例题以及问室友问了两天多，才开始理解一些。

至于题目，我真的没什么思路，后面的每道题都本想着尝试自己写出来，但基本都在试了2个小时左右后失败了，最后靠着ai一点一点理解做法。虽然收获很大，但这是我开学以来第一次感到学编程力不从心了：（