1. (15min)

sy119: 汉诺塔

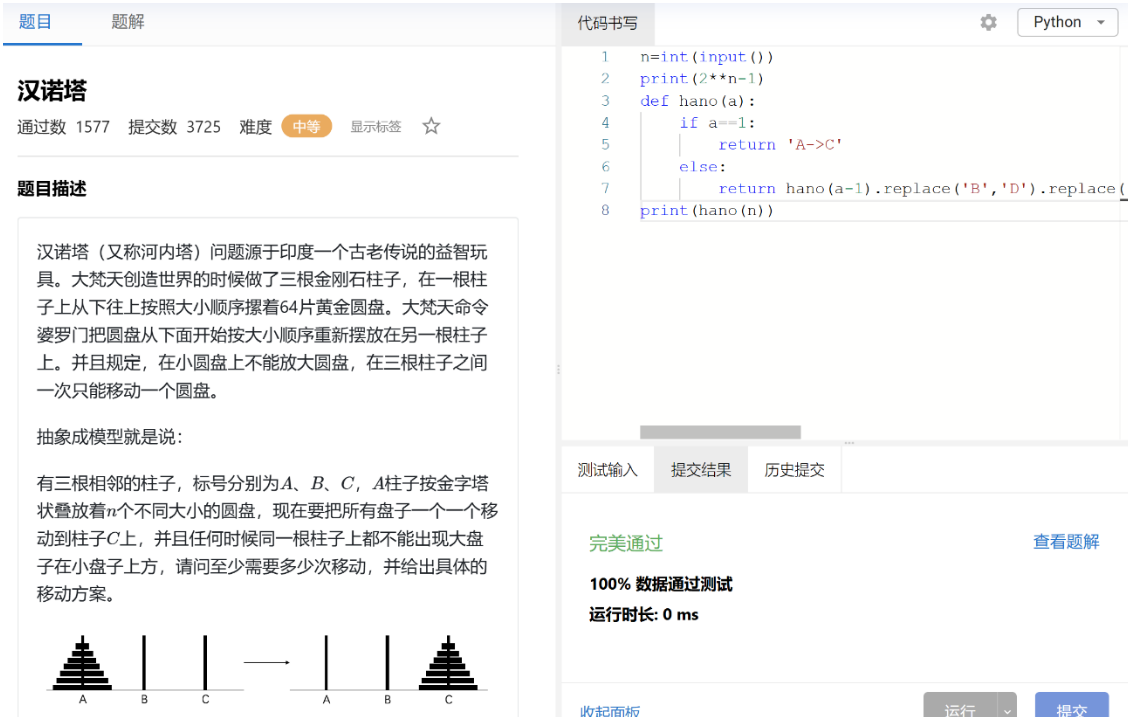
recursion, https://sunnywhy.com/sfbj/4/3/119

思路：用前一个 n 的步骤推出下一个的

代码：

```python
n=int(input())
print(2**n-1)
def hano(a):
    if a==1:
        return 'A->C'
    else:
        return hano(a-1).replace('B','D').replace('C','B').replace('D','C')+'\nA->C\n'+hano(a-1).replace('B','D').replace('A','B').replace('D','A')
print(hano(n))
```

运行：



汉诺塔

通过数 1577 提交数 3725 难度 中等 显示标签 ☆

题目描述

汉诺塔（又称河内塔）问题源于印度一个古老传说的益智玩具。大梵天创造世界的时候做了三根金刚石柱子，在一根柱子上从下往上按照大小顺序摞着64片黄金圆盘。大梵天命令婆罗门把圆盘从下面开始按大小顺序重新摆放在另一根柱子上。并且规定，在小圆盘上不能放大圆盘，在三根柱子之间一次只能移动一个圆盘。

抽象成模型就是说：

有三根相邻的柱子，标号分别为A、B、C，A柱子按金字塔状叠放着n个不同大小的圆盘，现在要把所有盘子一个一个移动到柱子C上，并且任何时候同一根柱子上都不能出现大盘子在小盘子上方，请问至少需要多少次移动，并给出具体的移动方案。

```python
n=int(input())
print(2**n-1)
def hano(a):
    if a==1:
        return 'A->C'
    else:
        return hano(a-1).replace('B','D').replace(
print(hano(n))
```

完美通过

100% 数据通过测试

运行时长: 0 ms

2. (20min)

sy132：全排列 I

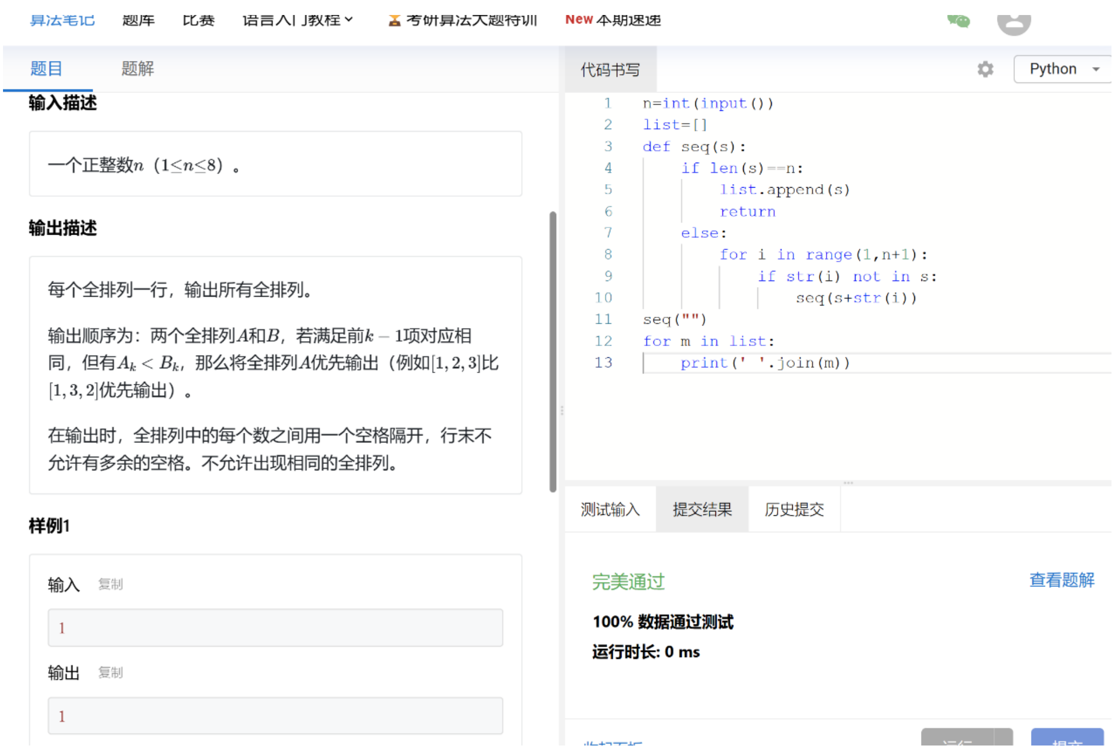
recursion, https://sunnywhy.com/sfbj/4/3/132

思路：还是用递归，函数递归的顺序自动实现排序

代码：

```
n=int(input())
list=[]
def seq(s):
    if len(s)==n:
        list.append(s)
        return
    else:
        for i in range(1,n+1):
            if str(i) not in s:
                seq(s+str(i))
seq("")
for m in list:
    print(' '.join(m))
```

运行：



输入描述

一个正整数n（$1 \le n \le 8$）。

输出描述

每个全排列一行，输出所有全排列。

输出顺序为：两个全排列A和B，若满足前$k-1$项对应相同，但有$A_k < B_k$，那么将全排列A优先输出（例如$[1,2,3]$比$[1,3,2]$优先输出）。

在输出时，全排列中的每个数之间用一个空格隔开，行末不允许有多余的空格。不允许出现相同的全排列。

样例1

输入 复制

```
1
```

输出 复制

```
1
```

```
n=int(input())
list=[]
def seq(s):
    if len(s)==n:
        list.append(s)
        return
    else:
        for i in range(1,n+1):
            if str(i) not in s:
                seq(s+str(i))
seq("")
for m in list:
    print(' '.join(m))
```

完美通过

100% 数据通过测试

运行时长: 0 ms

3. (20min)

02945: 拦截导弹

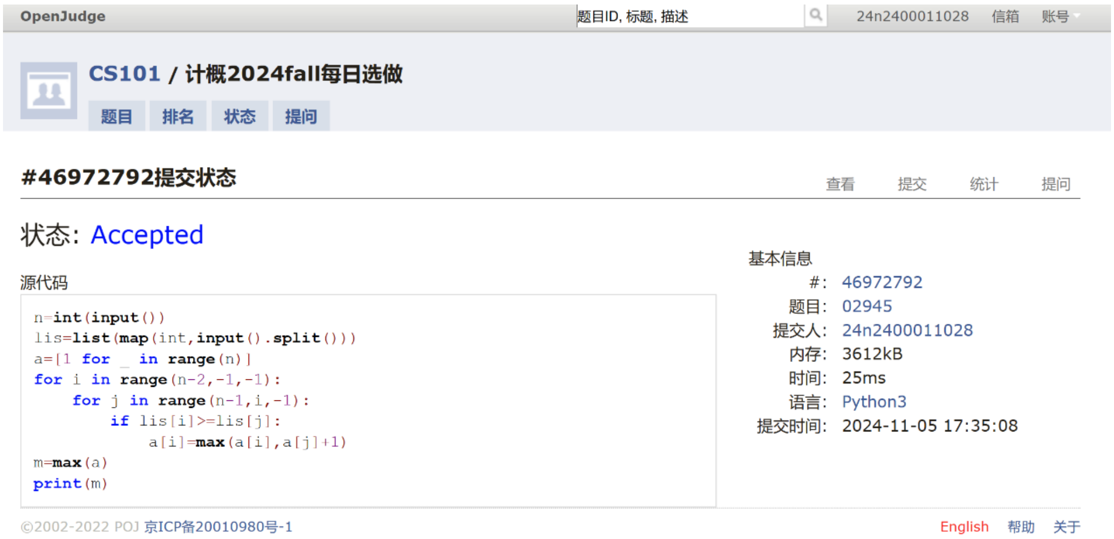
dp, http://cs101.openjudge.cn/2024fallroutine/02945

思路：从后往前找最长递减列

代码：

```
n=int(input())
lis=list(map(int,input().split()))
a=[1 for _ in range(n)]
for i in range(n-2,-1,-1):
    for j in range(n-1,i,-1):
        if lis[i]>=lis[j]:
            a[i]=max(a[i],a[j]+1)
m=max(a)
print(m)
```

运行：



#46972792提交状态

状态：Accepted

源代码

```
n=int(input())
lis=list(map(int,input().split()))
a=[1 for _ in range(n)]
for i in range(n-2,-1,-1):
    for j in range(n-1,i,-1):
        if lis[i]>=lis[j]:
            a[i]=max(a[i],a[j]+1)
m=max(a)
print(m)
```

基本信息

#：46972792
题目：02945
提交人：24n2400011028
内存：3612kB
时间：25ms
语言：Python3
提交时间：2024-11-05 17:35:08

4. (30min)

23421: 小偷背包

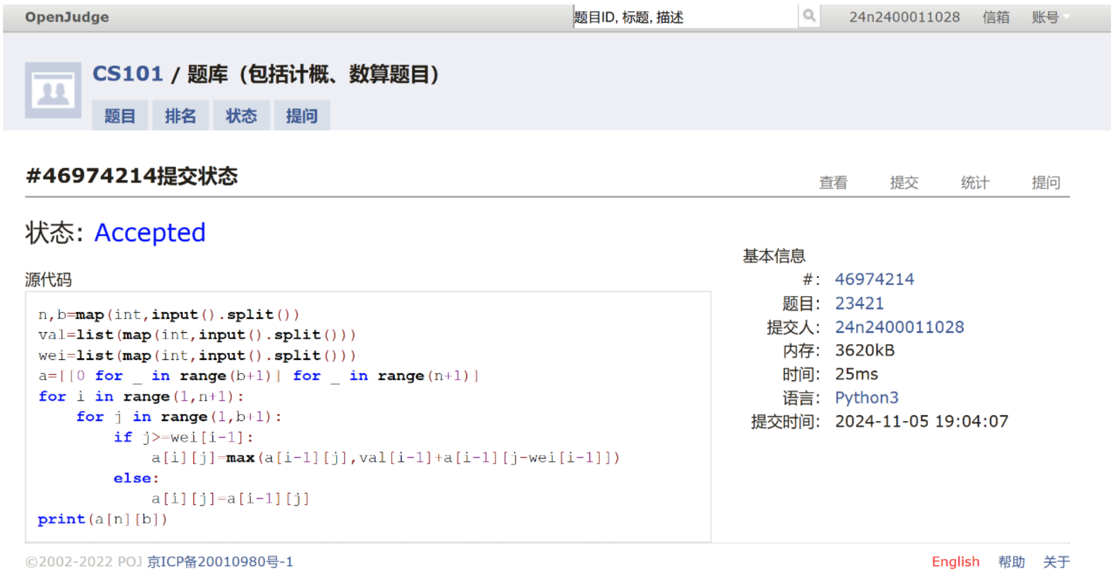
dp, http://cs101.openjudge.cn/practice/23421

思路：二维递归

代码：

```
n,b=map(int,input().split())
val=list(map(int,input().split()))
wei=list(map(int,input().split()))
a=[[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 j>=wei[i-1]:
            a[i][j]=max(a[i-1][j],val[i-1]+a[i-1][j-
wei[i-1]])
        else:
            a[i][j]=a[i-1][j]
print(a[n][b])
```

运行:



#46974214提交状态

状态: Accepted

源代码

```
n,b=map(int,input().split())
val=list(map(int,input().split()))
wei=list(map(int,input().split()))
a=[[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 j>=wei[i-1]:
            a[i][j]=max(a[i-1][j],val[i-1]+a[i-1][j-wei[i-1]])
        else:
            a[i][j]=a[i-1][j]
print(a[n][b])
```

基本信息

#: 46974214
题目: 23421
提交人: 24n2400011028
内存: 3620kB
时间: 25ms
语言: Python3
提交时间: 2024-11-05 19:04:07

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

5. (40min)

02754: 八皇后

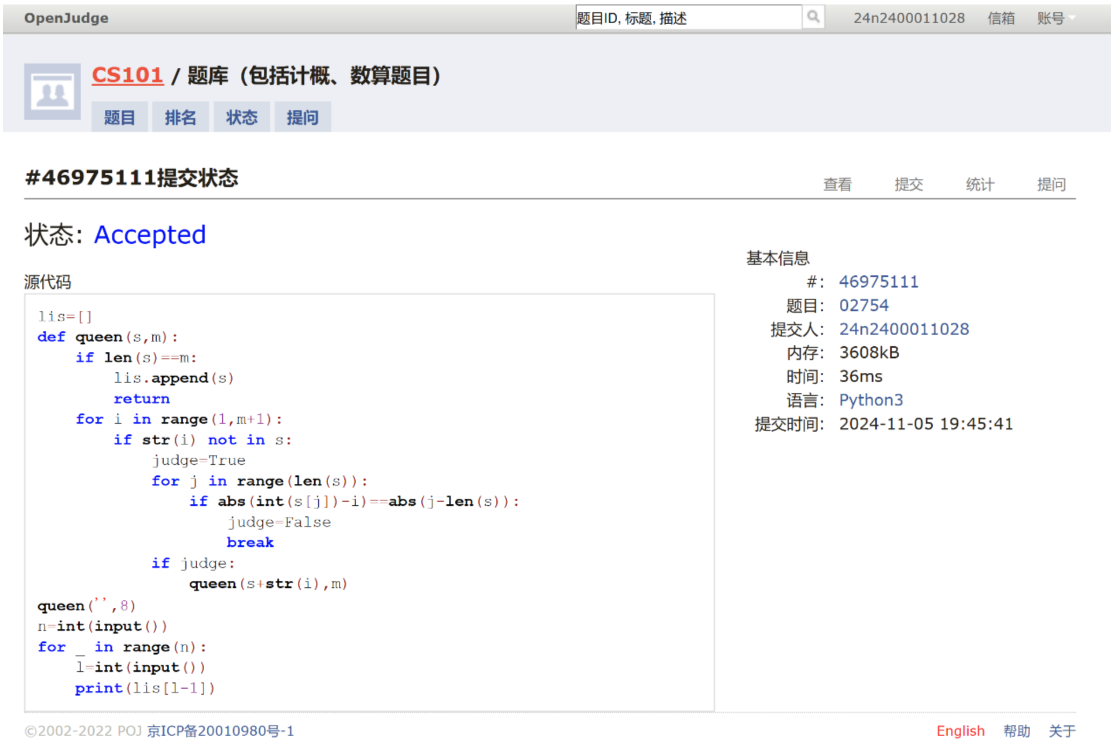
dfs and similar, http://cs101.openjudge.cn/practice/02754

思路: 与全排列类似

代码:

```
lis=[]
def queen(s,m):
    if len(s)==m:
        lis.append(s)
        return
    for i in range(1,m+1):
        if str(i) not in s:
            judge=True
            for j in range(len(s)):
                if abs(int(s[j])-i)==abs(j-len(s)):
                    judge=False
                    break
            if judge:
                queen(s+str(i),m)
queen('',8)
n=int(input())
for _ in range(n):
    l=int(input())
    print(lis[l-1])
```

运行：



#46975111提交状态

状态: Accepted

源代码

```
lis=[]
def queen(s,m):
    if len(s)==m:
        lis.append(s)
        return
    for i in range(1,m+1):
        if str(i) not in s:
            judge=True
            for j in range(len(s)):
                if abs(int(s[j])-i)==abs(j-len(s)):
                    judge=False
                    break
            if judge:
                queen(s+str(i),m)
queen('',8)
n=int(input())
for _ in range(n):
    l=int(input())
    print(lis[l-1])
```

基本信息

#:	46975111
题目:	02754
提交人:	24n2400011028
内存:	3608kB
时间:	36ms
语言:	Python3
提交时间:	2024-11-05 19:45:41

©2002-2022 POJ 京ICP备20010980号-1　English　帮助　关于

6. (20min)

189A. Cut Ribbon

brute force, dp 1300

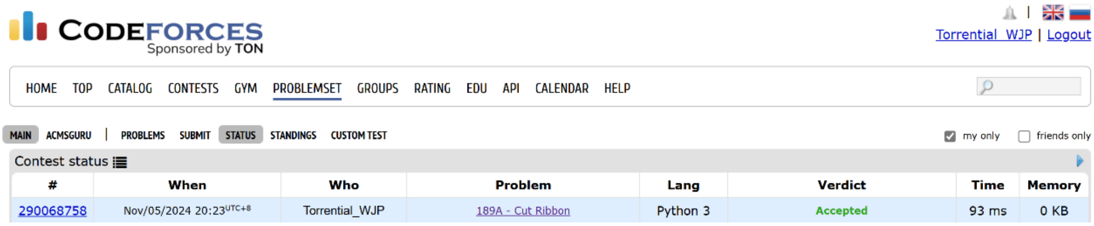
https://codeforces.com/problemset/problem/189/A

思路： 从小往大 dp

代码：

```
n,a,b,c=(map(int,input().split()))
lis=[0]*4001
lis[a]=1
lis[b]=1
lis[c]=1
for i in range(1,n+1):
    if i>=a and lis[i-a]!=0:
          lis[i]=max(lis[i],lis[i-a]+1)
    if i>=b and lis[i-b]!=0:
          lis[i]=max(lis[i],lis[i-b]+1)
    if i>=c and lis[i-c]!=0:
          lis[i]=max(lis[i],lis[i-c]+1)
print(lis[n])
```

运行：


CODEFORCES
Sponsored by TON
Torrential_WJP | Logout
HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP
MAIN ACMSGURU | PROBLEMS SUBMIT STATUS STANDINGS CUSTOM TEST
☑ my only □ friends only
Contest status

#	When	Who	Problem	Lang	Verdict	Time	Memory
290068758	Nov/05/2024 20:23^{UTC+8}	Torrential_WJP	189A - Cut Ribbon	Python 3	Accepted	93 ms	0 KB

总结和收获：

感觉到一维的 dp 感觉还是比较好想的，二维甚至更高维的 dp 就需要比较严谨的逻辑，不然容易出现疏漏