# Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Complied by 李振硕、信息管理系

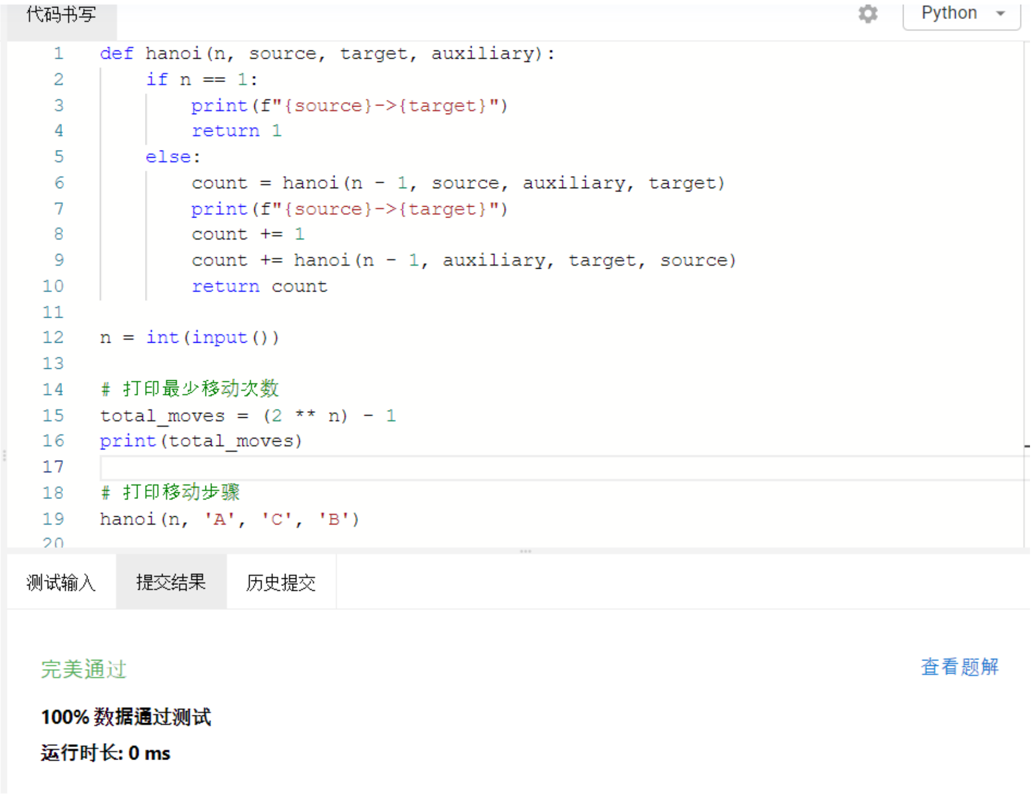
## 1. 题目

### sy119: 汉诺塔

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

思路：

代码：



```python
def hanoi(n, source, target, auxiliary):
    if n == 1:
        print(f"{source}->{target}")
        return 1
    else:
        count = hanoi(n - 1, source, auxiliary, target)
        print(f"{source}->{target}")
        count += 1
        count += hanoi(n - 1, auxiliary, target, source)
        return count

n = int(input())

# 打印最少移动次数
total_moves = (2 ** n) - 1
print(total_moves)

# 打印移动步骤
hanoi(n, 'A', 'C', 'B')
```

完美通过

100% 数据通过测试

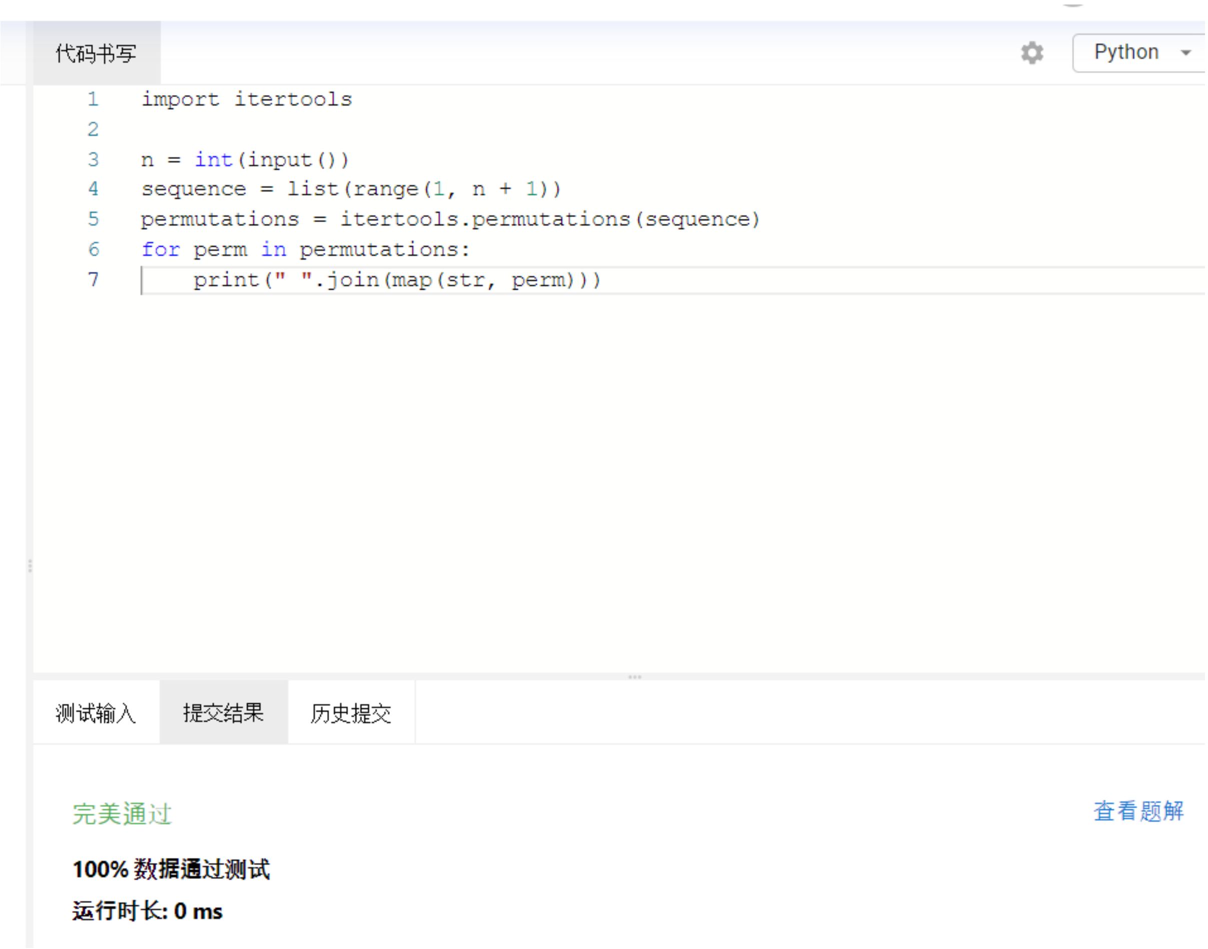
运行时长: 0 ms

### sy132: 全排列I

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

思路：

代码：



```python
import itertools

n = int(input())
sequence = list(range(1, n + 1))
permutations = itertools.permutations(sequence)
for perm in permutations:
    print(" ".join(map(str, perm)))
```

完美通过

100% 数据通过测试

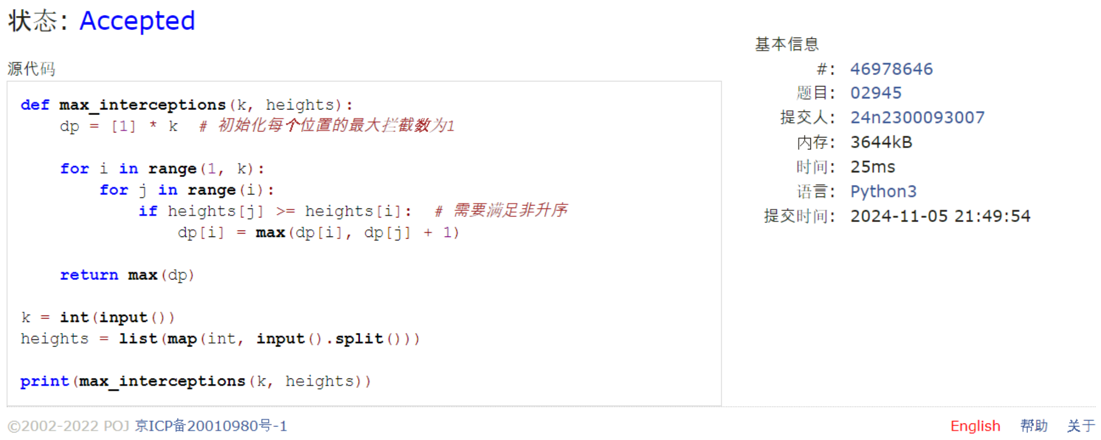
运行时长: 0 ms

### 02945: 拦截导弹

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

思路：

代码：



状态: Accepted

源代码

```python
def max_interceptions(k, heights):
    dp = [1] * k  # 初始化每个位置的最大拦截数为1

    for i in range(1, k):
        for j in range(i):
            if heights[j] >= heights[i]:  # 需要满足非升序
                dp[i] = max(dp[i], dp[j] + 1)

    return max(dp)

k = int(input())
heights = list(map(int, input().split()))

print(max_interceptions(k, heights))
```

基本信息

#:	46978646
题目:	02945
提交人:	24n2300093007
内存:	3644kB
时间:	25ms
语言:	Python3
提交时间:	2024-11-05 21:49:54

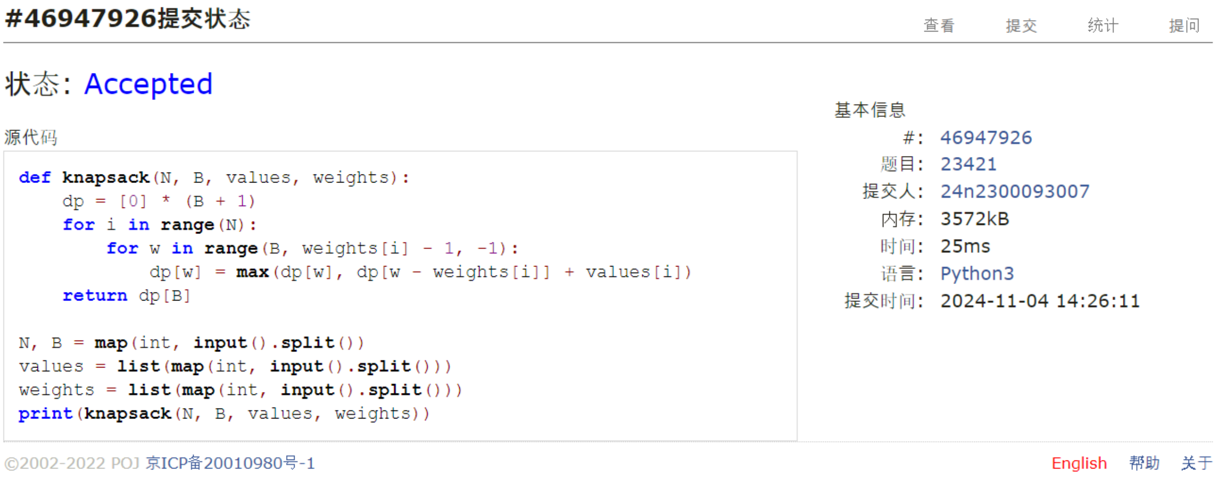
©2002-2022 POJ 京ICP备20010980号-1

### 23421: 小偷背包

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

思路：

代码：



#46947926提交状态

状态: Accepted

源代码

```
def knapsack(N, B, values, weights):
    dp = [0] * (B + 1)
    for i in range(N):
        for w in range(B, weights[i] - 1, -1):
            dp[w] = max(dp[w], dp[w - weights[i]] + values[i])
    return dp[B]

N, B = map(int, input().split())
values = list(map(int, input().split()))
weights = list(map(int, input().split()))
print(knapsack(N, B, values, weights))
```

基本信息

#:	46947926
题目:	23421
提交人:	24n2300093007
内存:	3572kB
时间:	25ms
语言:	Python3
提交时间:	2024-11-04 14:26:11

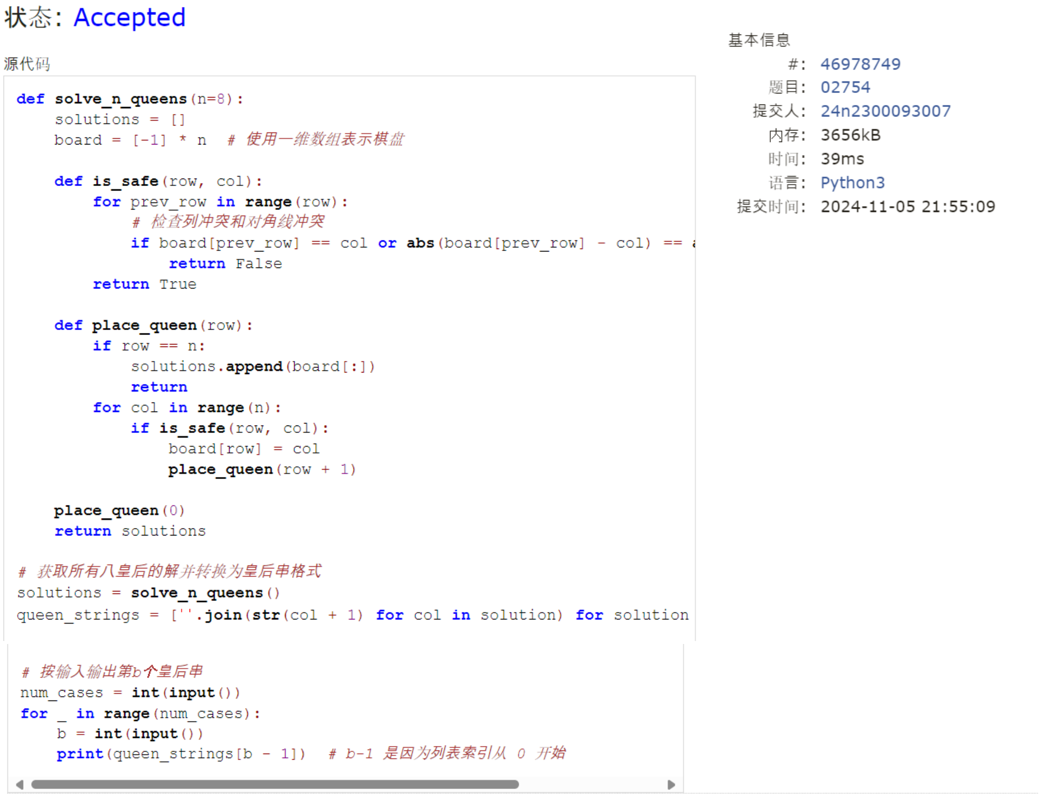
©2002-2022 POJ 京ICP备20010980号-1

### 02754: 八皇后

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

思路：

代码：



状态：Accepted

源代码

```
def solve_n_queens(n=8):
    solutions = []
    board = [-1] * n  # 使用一维数组表示棋盘

    def is_safe(row, col):
        for prev_row in range(row):
            # 检查列冲突和对角线冲突
            if board[prev_row] == col or abs(board[prev_row] - col) == a
                return False
        return True

    def place_queen(row):
        if row == n:
            solutions.append(board[:])
            return
        for col in range(n):
            if is_safe(row, col):
                board[row] = col
                place_queen(row + 1)

    place_queen(0)
    return solutions

# 获取所有八皇后的解并转换为皇后串格式
solutions = solve_n_queens()
queen_strings = [''.join(str(col + 1) for col in solution) for solution

# 按输入输出第b个皇后串
num_cases = int(input())
for _ in range(num_cases):
    b = int(input())
    print(queen_strings[b - 1])  # b-1 是因为列表索引从 0 开始
```

基本信息

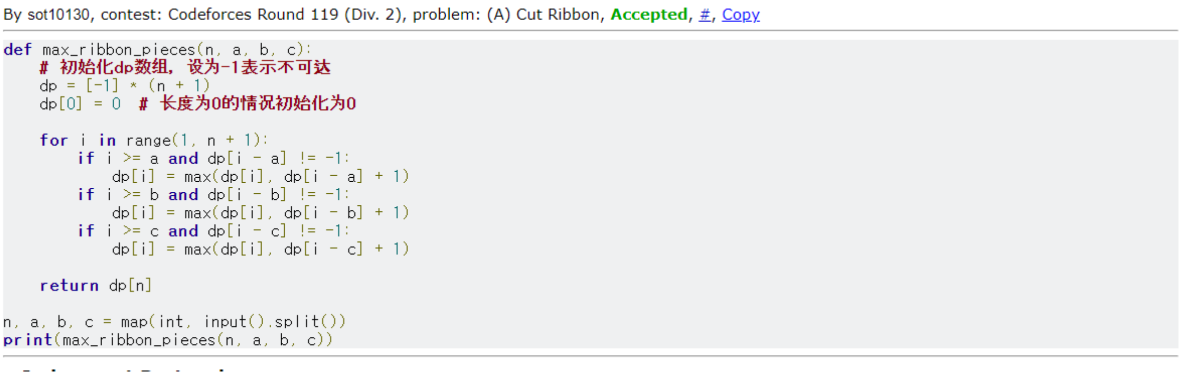
#：46978749
题目：02754
提交人：24n2300093007
内存：3656kB
时间：39ms
语言：Python3
提交时间：2024-11-05 21:55:09

### 189A. Cut Ribbon

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

花时：93ms

代码：



By sot10130, contest: Codeforces Round 119 (Div. 2), problem: (A) Cut Ribbon, Accepted, #, Copy

```
def max_ribbon_pieces(n, a, b, c):
    # 初始化dp数组, 设为-1表示不可达
    dp = [-1] * (n + 1)
    dp[0] = 0  # 长度为0的情况初始化为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)

    return dp[n]

n, a, b, c = map(int, input().split())
print(max_ribbon_pieces(n, a, b, c))
```

## 2. 学习总结和收获

这次作业感觉最难，很早之前开始写，现在才写完了。。我感到必须得重新复习DP RECURSION部分。