Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Complied by 佟永鑫 元培学院

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

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

思路：

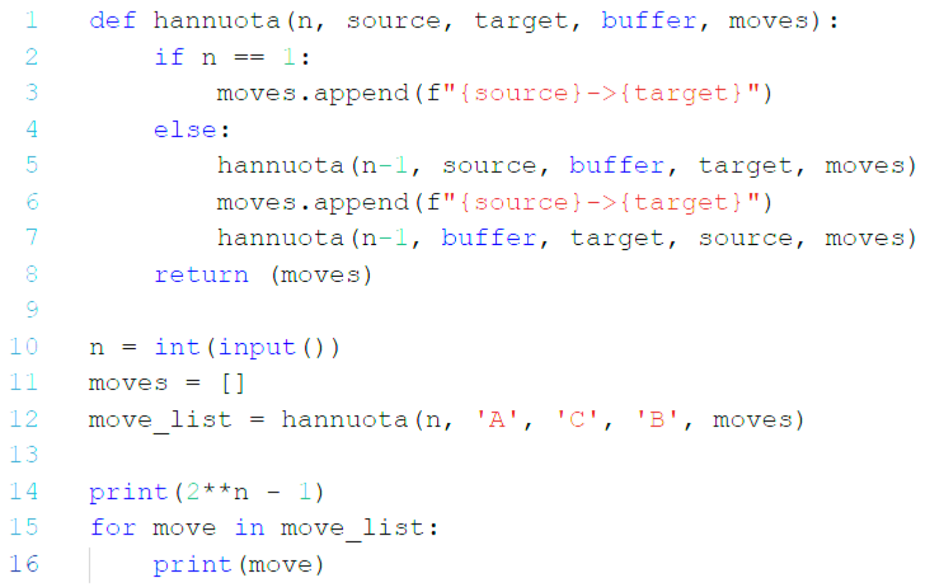
代码：

```python
def hannuota(n, source, target, buffer, moves):
    if n == 1:
        moves.append(f"{source}->{target}")
    else:
        hannuota(n-1, source, buffer, target, moves)
        moves.append(f"{source}->{target}")
        hannuota(n-1, buffer, target, source, moves)
    return (moves)

n = int(input())
moves = []
move_list = hannuota(n, 'A', 'C', 'B', moves)

print(2**n - 1)
for move in move_list:
    print(move)
```

代码运行截图 （至少包含有"Accepted"）



```
def hannuota(n, source, target, buffer, moves):
    if n == 1:
        moves.append(f"{source}->{target}")
    else:
        hannuota(n-1, source, buffer, target, moves)
        moves.append(f"{source}->{target}")
        hannuota(n-1, buffer, target, source, moves)
    return (moves)

n = int(input())
moves = []
move_list = hannuota(n, 'A', 'C', 'B', moves)

print(2**n - 1)
for move in move_list:
    print(move)
```

试输入
提交结果
历史提交
查看题解

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:

从小到大取出插入到新序列

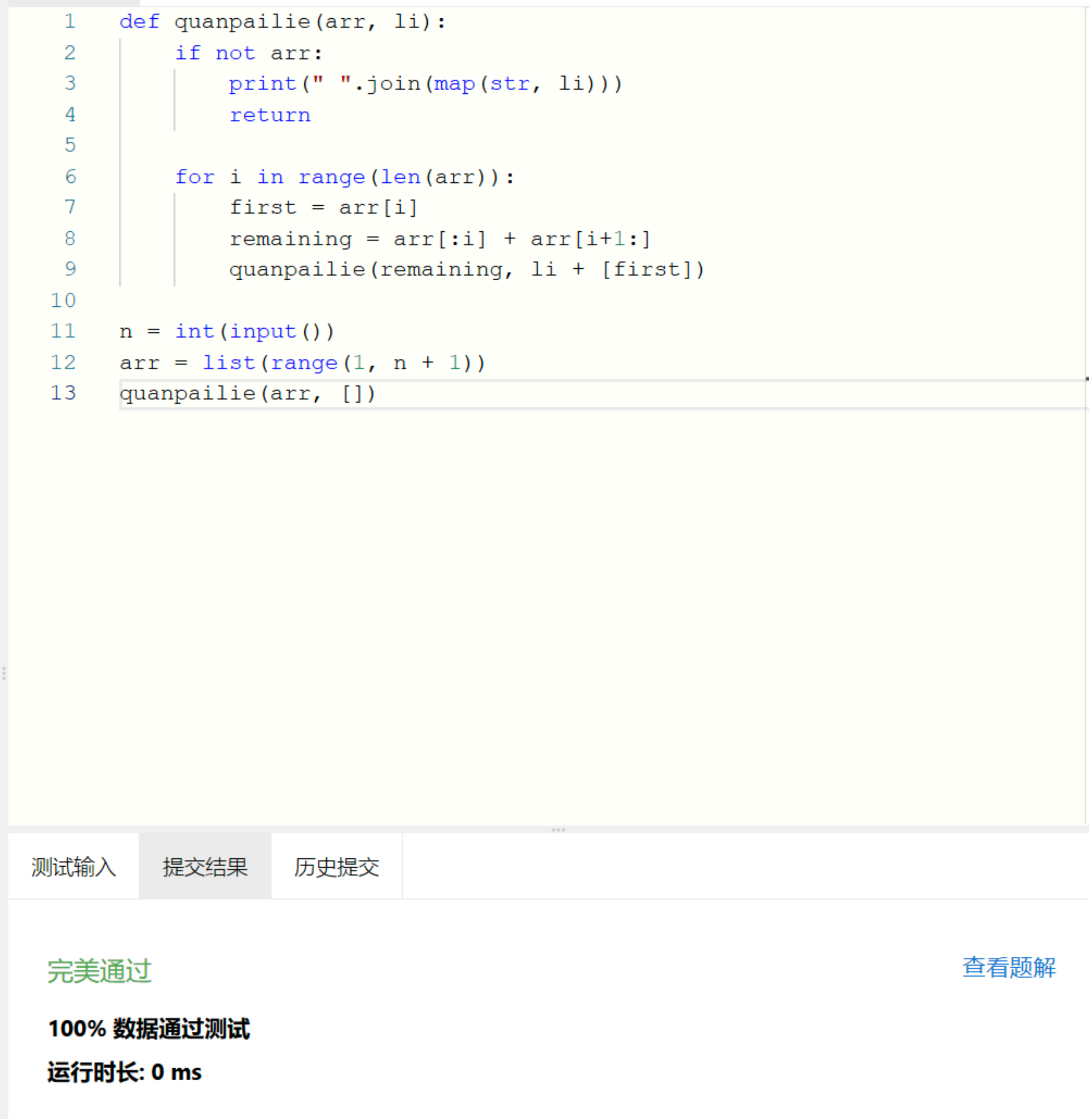
代码:

```python
def quanpailie(arr, li):
    if not arr:
        print(" ".join(map(str, li)))
        return

    for i in range(len(arr)):
        first = arr[i]
        remaining = arr[:i] + arr[i+1:]
        quanpailie(remaining, li + [first])

n = int(input())
arr = list(range(1, n + 1))
quanpailie(arr, [])
```

代码运行截图 == （至少包含有"Accepted"） ==



```python
def quanpailie(arr, li):
    if not arr:
        print(" ".join(map(str, li)))
        return

    for i in range(len(arr)):
        first = arr[i]
        remaining = arr[:i] + arr[i+1:]
        quanpailie(remaining, li + [first])

n = int(input())
arr = list(range(1, n + 1))
quanpailie(arr, [])
```

完美通过

100% 数据通过测试

运行时长: 0 ms

02945: 拦截导弹

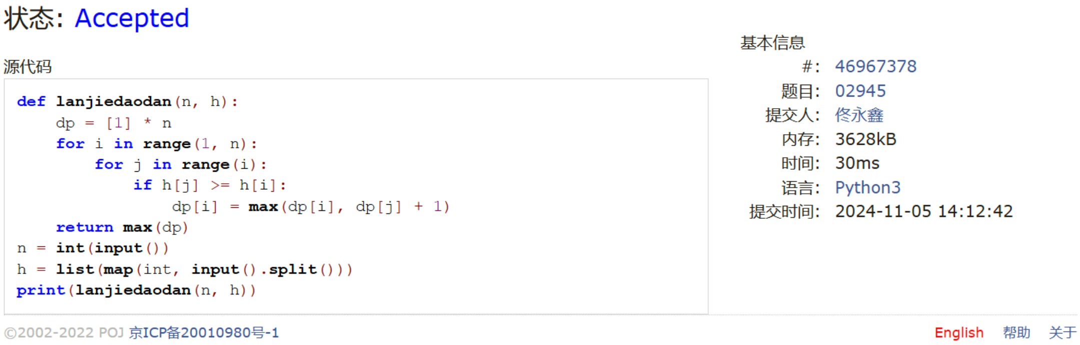
dp, http://cs101.openjudge.cn/2024fallroutine/02945

思路：

代码：

```python
def lanjiedaodan(n, h):
    dp = [1] * n
    for i in range(1, n):
        for j in range(i):
            if h[j] >= h[i]:
                dp[i] = max(dp[i], dp[j] + 1)
    return max(dp)
n = int(input())
h = list(map(int, input().split()))
print(lanjiedaodan(n, h))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
def lanjiedaodan(n, h):
    dp = [1] * n
    for i in range(1, n):
        for j in range(i):
            if h[j] >= h[i]:
                dp[i] = max(dp[i], dp[j] + 1)
    return max(dp)
n = int(input())
h = list(map(int, input().split()))
print(lanjiedaodan(n, h))
```

基本信息

#:	46967378
题目:	02945
提交人:	佟永鑫
内存:	3628kB
时间:	30ms
语言:	Python3
提交时间:	2024-11-05 14:12:42

©2002-2022 POJ 京ICP备20010980号-1

23421: 小偷背包

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

思路：

思路没有拦截导弹直接，参考了AI

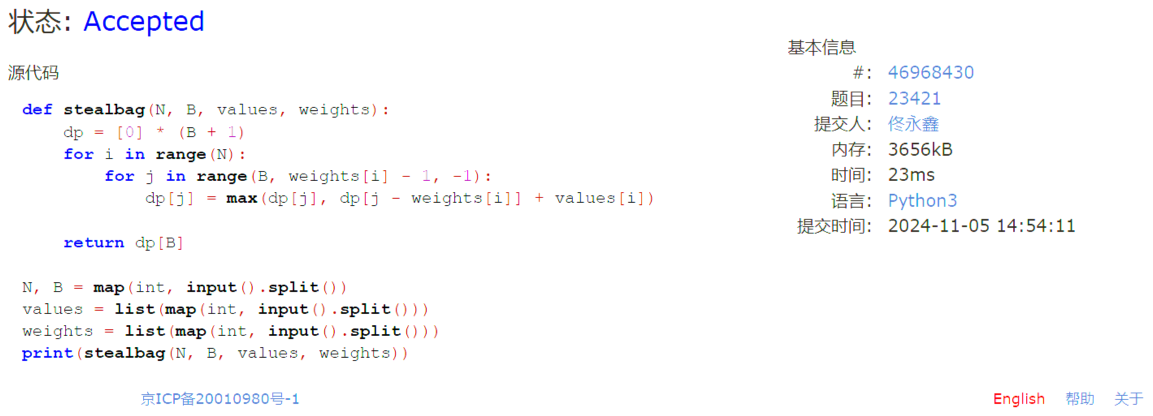
代码：

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

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

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

基本信息

#:	46968430
题目:	23421
提交人:	佟永鑫
内存:	3656kB
时间:	23ms
语言:	Python3
提交时间:	2024-11-05 14:54:11

02754: 八皇后

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

思路:

利用前题全排列算法加穷举检验，算法比较丑，后续再尝试递归方法

代码:

```python
def quanpailie(arr, li, result):
    if not arr:
        result.append(li)
        return

    for i in range(len(arr)):
        first = arr[i]
        remaining = arr[:i] + arr[i+1:]
        quanpailie(remaining, li + [first], result)

arr = list(range(1, 9))
result = []
quanpailie(arr, [], result)
queens = []
```

```
for i in range(len(result)):
    queen = result[i]
    skip2 = False
    for j in range(8):
        skip1 = False
        for k in range(j+1, 8):
            if k - j == abs(queen[k] - queen[j]):
                skip1 = True
                break
        if skip1:
            skip2 = True
    if not skip2:
        queens.append(queen)

n = int(input())
for _ in range(n):
    b = int(input())
    solution = queens[b-1]
    print(''.join(map(str, solution)))
```

代码运行截图 （至少包含有"Accepted"）

189A. Cut Ribbon

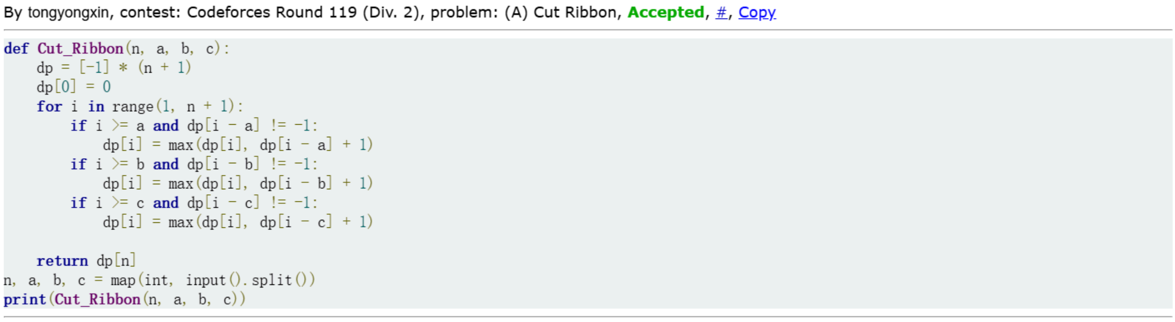
brute force, dp 1300 https://codeforces.com/problemset/problem/189/A

思路：

代码：

```
def Cut_Ribbon(n, a, b, c):
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)
return dp[n]
n, a, b, c = map(int, input().split())
print(Cut_Ribbon(n, a, b, c))
```

代码运行截图 （至少包含有"Accepted"）



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

```
def Cut_Ribbon(n, a, b, c):
    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)

    return dp[n]
n, a, b, c = map(int, input().split())
print(Cut_Ribbon(n, a, b, c))
```

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。

感觉递归和DP比贪心难，特别是递归，能感觉出大概是这样递归的，但把想法转换成代码还是感觉有些绕。