

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 徐贤天、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

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

思路:

用函数写出递推公式，每次把不同的值传入函数的变量中。把n个圆盘的移动分解为n-1个圆盘与最后一个

代码:

```
def move(n, from_, to_, mid_):  
    if n == 0:  
        return  
    else:  
        move(n-1, from_, mid_, to_)  
        print('{}->{}'.format(from_, to_))  
        move(n-1, mid_, to_, from_)  
n = int(input())  
print(2**n-1)  
move(n, 'A', 'C', 'B')
```

代码运行截图 (至少包含有"Accepted")

代码书写 Python

```
1 def move(n, from_, to_, mid_):
2     if n == 0:
3         return
4     else:
5         move(n-1, from_, mid_, to_)
6         print('{}->{}'.format(from_, to_))
7         move(n-1, mid_, to_, from_)
8 n = int(input())
9 print(2**n-1)
10 move(n, 'A', 'C', 'B')
```

测试输入 提交结果 历史提交

完美通过

100% 数据通过测试

运行时长: 0 ms

查看题解

收起面板 运行 提交

sy132: 全排列I

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

思路:

注意要用seq[:]创建浅拷贝再加入ret中, 否则seq会变化

代码:

```

def qpl(idx,n,used,seq,ret):
    if idx == n+1:
        ret.append(seq[:])
        return
    else:
        for i in range(1,n+1):
            if not used[i]:
                seq.append(str(i))
                used[i] = True
                qpl(idx+1,n,used,seq,ret)
                used[i] = False
                seq.pop()

idx = 1
n = int(input())
used = [False]*(n+1)
seq = []
ret = []
qpl(idx,n,used,seq,ret)
for i in ret:
    print(' '.join(i))

```

代码运行截图 == (至少包含有"Accepted") ==

代码书写



Python ▾

```
1 def qpl(idx,n,used,seq,ret):
2     if idx == n+1:
3         ret.append(seq[:])
4         return
5     else:
6         for i in range(1,n+1):
7             if not used[i]:
8                 seq.append(str(i))
9                 used[i] = True
10                qpl(idx+1,n,used,seq,ret)
11                used[i] = False
12                seq.pop()
13
14     idx = 1
15     n = int(input())
16     used = [False]*(n+1)
17     seq = []
18     ret = []
19     qpl(idx,n,used,seq,ret)
20     for i in ret:
21         print(' '.join(i))
```

测试输入

提交结果

历史提交

完美通过

100% 数据通过测试

运行时长: 0 ms

查看题解



收起面板

运行 ▾

提交

02945: 拦截导弹

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

思路:

用memo字典储存已经计算过的、从第i个数字开始的最大递减序列长度, 节省时间

代码:

```

memo = {}
def L(heights,i):
    if i in memo:
        return memo[i]

    if i == k - 1:
        return 1

    max_len = 1
    for j in range(i+1,len(heights)):
        if heights[j] <= heights[i]:
            max_len = max(max_len,L(heights,j)+1)

    memo[i] = max_len
    return max_len

k = int(input())
heights = list(map(int,input().split()))
ret = [L(heights,i) for i in range(k)]
print(max(ret))

```

代码运行截图 (至少包含有"Accepted")

#46930604提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

memo = {}
def L(heights,i):
    if i in memo:
        return memo[i]

    if i == k - 1:
        return 1

    max_len = 1
    for j in range(i+1,len(heights)):
        if heights[j] <= heights[i]:
            max_len = max(max_len,L(heights,j)+1)

    memo[i] = max_len
    return max_len

k = int(input())
heights = list(map(int,input().split()))
ret = [L(heights,i) for i in range(k)]
print(max(ret))

```

基本信息

#: 46930604
 题目: 02945
 提交人: 24n2400011033
 内存: 3652kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-11-03 16:34:22

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

23421: 小偷背包

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

思路:

代码:

```
def max_v(i, left_weight):
    if left_weight < 0:
        return -float('inf')
    if i == N:
        return 0
    return max(max_v(i+1, left_weight), v[i] + max_v(i+1, left_weight-w[i]))

N, B = map(int, input().split())
v = list(map(int, input().split()))
w = list(map(int, input().split()))
print(max_v(0, B))
```

代码运行截图 (至少包含有"Accepted")

#46932081提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def max_v(i, left_weight):
    if left_weight < 0:
        return -float('inf')
    if i == N:
        return 0
    return max(max_v(i+1, left_weight), v[i] + max_v(i+1, left_weight-w[i]))

N, B = map(int, input().split())
v = list(map(int, input().split()))
w = list(map(int, input().split()))
print(max_v(0, B))
```

基本信息

#: 46932081
 题目: 23421
 提交人: 24n2400011033
 内存: 3572kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-11-03 17:06:52

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

02754: 八皇后

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

思路:

代码:

```
ans = []
def Queens(seq):
    for j in range(1,9): #在这一行遍历1~8列, j为列数
        for x in range(len(seq)): #找出已经存在的皇后, x为皇后所在行, int(seq[x])为皇后所在列
            if seq[x] == str(j) or abs(j-int(seq[x])) == abs(len(seq)-x): #如果在同一列or行间距=列间距
                break #判断第j列能否摆放皇后
            else: #若j对于每个已存在的皇后都不能被吃
                if len(seq) == 7: #如果已经到了最后一行
                    ans.append(seq+str(j)) #将s+str(j)压入ans中
                    Queens(seq+str(j)) #进入下一层
Queens('') #生成ans
```

```
n = int(input())
for _ in range(n):
    b = int(input())
    print(ans[b-1])
```

代码运行截图 (至少包含有"Accepted")

#46941437提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
ans = []
def Queens(seq):
    for j in range(1,9): #在这一行遍历1~8列, j为列数
        for x in range(len(seq)): #找出已经存在的皇后, x为皇后所在行, int(seq[x])为皇后所在列
            if seq[x] == str(j) or abs(j-int(seq[x])) == abs(len(seq)-x):
                break #判断第j列能否摆放皇后
            else: #若j对于每个已存在的皇后都不能被吃
                if len(seq) == 7: #如果已经到了最后一行
                    ans.append(seq+str(j)) #将s+str(j)压入ans中
                Queens(seq+str(j)) #进入下一层
Queens('') #生成ans

n = int(input())
for _ in range(n):
    b = int(input())
    print(ans[b-1])
```

基本信息

#: 46941437
题目: 02754
提交人: 24n2400011033
内存: 3632kB
时间: 48ms
语言: Python3
提交时间: 2024-11-03 23:07:23

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

189A. Cut Ribbon

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

思路:

代码:

```
n, a, b, c = map(int, input().split())
dp = [0] + [-float('inf')] * 4000
for i in range(1, n+1): #i为对应长度的ribbon
    dp[i] = max(dp[i-a], dp[i-b], dp[i-c]) + 1 #局部最优
print(dp[n])
```

代码运行截图 (至少包含有"Accepted")

By Aunixt, contest: Codeforces Round 119 (Div. 2), problem: (A) Cut Ribbon, **Accepted**, #, [Copy](#)

```
n, a, b, c = map(int, input().split())
dp = [0] + [-float('inf')] * 4000
for i in range(1, n+1): #i为对应长度的ribbon
    dp[i] = max(dp[i-a], dp[i-b], dp[i-c]) + 1 #局部最优
print(dp[n])
```

 解释 

→Judgement Protocol

Test: #1, time: 46 ms., memory: 16 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

5 5 3 2

Output

2

Answer

2

Checker Log

ok 2

Test: #2, time: 46 ms., memory: 4 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

7 5 5 2

Output

2

Answer

2

Checker Log

ok 2

Test: #3, time: 62 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

4 4 4 4

Output

1

Answer

1

Checker Log

ok 1

Test: #4, time: 61 ms., memory: 8 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

1 1 1 1

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。

加上了递归和dp之后明显感觉题目的思维量变得更大了，可能是对这些算法还不够熟悉的缘故吧。这次作业很多题目都没有思路，只能对着题解一段代码一段代码地看，再借助ai和pythontutor的辅助，总算把递归和dp的思路理得更清楚一些了，之后再凭着印象自己打一遍。此外看了B站上的网课、csdn和知乎上的一些文章，来加深理解。

期中加油！