

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 任宇桐 物理学院

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:

将 n 个积木移动到C柱子上, 相当于将 $(n-1)$ 个积木移动到B柱子上。主要学习到了如何声明全局变量, 最后进行操作。

代码:

```
n = int(input())
cnt = 0
ans = []
def f(n, x, y, z):
    global cnt
    if n == 1:
        ans.append(f'{x}->{z}')
        cnt += 1
    else:
        f(n-1, x, z, y)
        ans.append(f'{x}->{z}')
        cnt += 1
        f(n-1, y, x, z)

f(n, 'A', 'B', 'C')
print(cnt)
print(*ans, sep = '\n')
```

代码运行截图 (至少包含有"Accepted")

题目

题解

汉诺塔

通过数 1421 提交数 3473 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，并且任何时刻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。



代码书写

Python

```
1 n = int(input())
2 cnt = 0
3 ans = []
4 def f(n, x, y, z):
5     global cnt
6     if n == 1:
7         ans.append(f'{x}->{z}')
8         cnt += 1
9     else:
10        f(n-1, x, z, y)
11        ans.append(f'{x}->{z}')
12        cnt += 1
13        f(n-1, y, x, z)
14
15 f(n, 'A', 'B', 'C')
16 print(cnt)
17 print(*ans, sep = '\n')
```

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

收起面板

运行

提交

sy132: 全排列I

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

思路：

之前仅仅想到了递推的思想，通过阅读题解学会了使用回溯的方法处理，不需要进行排序了。

代码：

```
# 自己写的代码
n = int(input())
def f(n):
    if n == 1:
        return [[1]]
    else:
        new = []
        pre = f(n-1)
        for i in pre:
            for j in range(n):
                s = i[:]
                s.insert(j, n)
                new.append(s)
        return new

ans = f(n)
ans.sort()
for k in ans:
    print(*k, sep = ' ')

# 看了题解以后学会的代码
n = int(input())
def f(n, idx, used, buffer, ans):
    if idx == n+1:
        ans.append(buffer[:])
        return
```

```

    for i in range(1, n+1):
        if not used[i]:
            buffer.append(i)
            used[i] = True
            f(n, idx+1, used, buffer, ans)
            used[i] = False
            buffer.pop()

idx = 1
used = [False]*(n+1)
buffer = []
ans = []
f(n, idx, used, buffer, ans)
for i in ans:
    print(*i)

```

代码运行截图 (至少包含有"Accepted")

全排列

通过数 1027 提交数 2171 难度 中等 显示标签 ☆

题目描述

给定一个正整数 n ，假设序列 $S = [1, 2, 3, \dots, n]$ ，求 S 的全排列。

输入描述

一个正整数 n ($1 \leq n \leq 8$)。

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。

```

1  n = int(input())
2  def f(n, idx, used, buffer, ans):
3      if idx == n+1:
4          ans.append(buffer[:])
5          return
6      for i in range(1, n+1):
7          if not used[i]:
8              buffer.append(i)
9              used[i] = True
10             f(n, idx+1, used, buffer, ans)
11             used[i] = False
12             buffer.pop()
13
14  idx = 1
15  used = [False]*(n+1)
16  buffer = []
17  ans = []

```

测试输入 提交结果 历史提交

完美通过 查看题解

100% 数据通过测试
运行时长: 0 ms

收起面板 运行 提交

02945: 拦截导弹

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

思路：

思路还是比较明确的，但是注意初始化的赋值，一开始全部赋值成了0，发现是错误的，可以自己先想一些corner case 判断一下初值是否正确。

代码：

```

k = int(input())
s = list(map(int, input().split()))
dp = [1]*k
for i in range(k):
    for j in range(i):
        if s[j] >= s[i]:
            dp[i] = max(dp[i], dp[j]+1)
print(max(dp))

```

代码运行截图 (至少包含有"Accepted")

#46757510提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

k = int(input())
s = list(map(int, input().split()))
dp = [1]*k
for i in range(k):
    for j in range(i):
        if s[j] >= s[i]:
            dp[i] = max(dp[i], dp[j]+1)
print(max(dp))

```

基本信息

#: 46757510
题目: 02945
提交人: 24n2400011498
内存: 3596kB
时间: 29ms
语言: Python3
提交时间: 2024-10-26 19:04:48

©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:

感觉第一次完成这个题目还是有一点难度的, 但是见一次就应该容易掌握了。

代码:

```

n, b = map(int, input().split())
s = list(map(int, input().split()))
w = list(map(int, input().split()))
dp = [[0] * (b+1) for _ in range(n+1)]
for i in range(1, n+1):
    for j in range(1, b+1):
        if j >= w[i-1]:
            dp[i][j] = max(dp[i-1][j], dp[i-1][j-w[i-1]]+s[i-1])
        else:
            dp[i][j] = dp[i-1][j]
print(dp[-1][-1])

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

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

基本信息

#: 46812667
题目: 23421
提交人: 24n2400011498
内存: 3648kB
时间: 24ms
语言: Python3
提交时间: 2024-10-29 15:30:15

02754: 八皇后

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

思路:

因为先完成了全排列的问题, 发现再次做8皇后的问题变得简单了不少, 按照顺序递推求解即可。

代码:

```
def f(buffer, idx, ans):
    if idx == 8:
        ans.append(buffer[:])
        return
    for i in range(1, 9):
        if all(buffer[k] != i and abs(idx-k) != abs(i-buffer[k]) for k in
range(len(buffer))):
            buffer.append(i)
            f(buffer, idx+1, ans)
            buffer.pop()

ans = []
idx = 0
buffer = []
f(buffer, idx, ans)
n = int(input())
for _ in range(n):
    s = int(input())
    print(*ans[s-1], sep = '')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
def f(buffer, idx, ans):
    if idx == 8:
        ans.append(buffer[:])
        return
    for i in range(1,9):
        if all(buffer[k] != i and abs(idx-k) != abs(i-buffer[k])) for k in range(1,9):
            buffer.append(i)
            f(buffer, idx+1, ans)
            buffer.pop()

ans = []
idx = 0
buffer = []
f(buffer, idx, ans)
n = int(input())
for _ in range(n):
    s = int(input())
    print(*ans[s-1], sep = '')
```

基本信息

#: 46839130
 题目: 02754
 提交人: 24n2400011498
 内存: 3576kB
 时间: 40ms
 语言: Python3
 提交时间: 2024-10-30 18:00:47

189A. Cut Ribbon

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

思路:

第一次完成这个问题还是用了不少功夫的，因为不知道如何处理长度不足a,b,c时的递推。然后参考了题解以后发现原来还能这么简单。

代码:

```
# 原来的代码
n, a, b, c = map(int, (input().split()))
s = [a,b,c]
s.sort()
a = s[0]
b = s[1]
c = s[2]
dp = [0] *(n+1)
try:
    dp[a] = 1
    for i in range(a + 1, b + 1):
        dp[i] = dp[i - a] + (1 if dp[i - a] != 0 else 0)
    dp[b] = max(dp[b], 1)
    for i in range(b + 1, c + 1):
        dp[i] = max(dp[i - a] + (1 if dp[i - a] != 0 else 0), dp[i - b] + (1 if
dp[i - b] != 0 else 0))
    dp[c] = max(dp[c], 1)
    for i in range(c + 1, n + 1):
        dp[i] = max(dp[i - a] + (1 if dp[i - a] != 0 else 0), dp[i - b] + (1 if
dp[i - b] != 0 else 0),
                    dp[i - c] + (1 if dp[i - c] != 0 else 0))
    print(dp[-1])
except IndexError:
    print(dp[-1])
```

代码运行截图 (至少包含有"Accepted")

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
288213110	Practice: stur	189A - 39	Python 3	Accepted	77 ms	44 KB	2024-10-27 05:30:09	2024-10-27 05:30:09	★	Compare

→ Source

Copy

```
n, a, b, c = map(int, (input().split()))
s = [a, b, c]
s.sort()
a = s[0]
b = s[1]
c = s[2]
dp = [0] * (n+1)
try:
    dp[a] = 1
    for i in range(a+1, b+1):
        dp[i] = dp[i-a] + (1 if dp[i-a] != 0 else 0)
    dp[b] = max(dp[b], 1)
    for i in range(b+1, c+1):
        dp[i] = max(dp[i-a] + (1 if dp[i-a] != 0 else 0), dp[i-b] + (1 if dp[i-b] != 0 else 0))
    dp[c] = max(dp[c], 1)
    for i in range(c+1, n+1):
        dp[i] = max(dp[i-a] + (1 if dp[i-a] != 0 else 0), dp[i-b] + (1 if dp[i-b] != 0 else 0),
                    dp[i-c] + (1 if dp[i-c] != 0 else 0))
    print(dp[-1])
except IndexError:
    print(dp[-1])
```

[Click](#) to see test details

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。

作业题目感觉变难了，感觉比较容易的思路还是用数学方法解决的问题（代码简单），学会了一些递归的表达式，发现dp问题最容易出错的就是初始赋值的问题，以后设置初值的时候要先想一下简单的样例，尽量一次找到正确的值。