

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 魏佳亮 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：递归，n是n-1的把B,C调换加上A->C加上A,B调换

代码：

```
import copy
n = int(input())
l = [[] for _ in range(16)]
l[0] = ['A->C']
for i in range(1,16):
    a = copy.deepcopy(l[i-1])
    b = copy.deepcopy(l[i-1])
    for j in range(2**(i)-1):
        a[j] = a[j].replace('B', 'O')
        a[j] = a[j].replace('C', 'B')
        a[j] = a[j].replace('O', 'C')
        b[j] = b[j].replace('A', 'O')
        b[j] = b[j].replace('B', 'A')
        b[j] = b[j].replace('O', 'B')
    l[i] = a+['A->C']+b
print(2**n-1)
print(*l[n-1],sep='\n')
```

代码运行截图 (至少包含有"Accepted")

汉诺塔

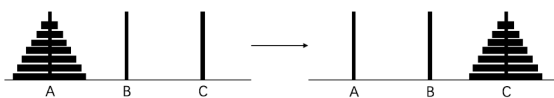
通过数 1421 提交数 3476 难度 中等 显示标签 ☆

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。



输入描述

一个正整数n ($1 \leq n \leq 16$)，表示圆盘的个数。

```
1 import copy
2 n = int(input())
3 l = [[] for _ in range(16)]
4 l[0] = ['A->C']
5 for i in range(1,16):
6     a = copy.deepcopy(l[i-1])
7     b = copy.deepcopy(l[i-1])
8     for j in range(2*(i)-1):
9         a[j] = a[j].replace('B','0')
10        a[j] = a[j].replace('C','B')
11        a[j] = a[j].replace('0','C')
12        b[j] = b[j].replace('A','0')
13        b[j] = b[j].replace('B','A')
14        b[j] = b[j].replace('0','B')
15    l[i] = a+['A->C']+b
16 print(2**n-1)
17 print(*l[n-1],sep='\n')
```

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

sy132: 全排列

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

思路：和上一题类似，只不过是把调换变成部分数+1

代码：

```
import copy
from math import factorial
n = int(input())
l = [[] for _ in range(8)]
l[0] = [[1]]
for i in range(1,8):
    for j in range(i+1):
        a = copy.deepcopy(l[i-1])
        for w in range(factorial(i)):
            for k in range(i):
                if a[w][k] > j:
                    a[w][k] += 1
            a[w] = [j+1]+a[w]
        l[i] = l[i]+a
print(*( ' '.join([str(y) for y in x]) for x in l[n-1]),sep='\n')
```

代码运行截图 == (至少包含有"Accepted") ==

题目

题解

全排列

通过数 1023 提交数 2165 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。

样例1

输入 复制

1

代码书写

Python

```
1 import copy
2 from math import factorial
3 n = int(input())
4 l = [[] for _ in range(8)]
5 l[0] = [[1]]
6 for i in range(1,8):
7     for j in range(i+1):
8         a = copy.deepcopy(l[i-1])
9         for w in range(factorial(i)):
10             for k in range(i):
11                 if a[w][k] > j:
12                     a[w][k] += 1
13             a[w] = [j+1]+a[w]
14             l[i] = l[i]+a
15 print(*( ' '.join([str(y) for y in x]) for x in l[n-1]),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

查看答案

收起面板

运行

提交

02945: 拦截导弹

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

思路：最大下降子序列

代码：

```
k = int(input())
l = list(map(int,input().split()))
t = [1 for _ in range(k)]
for i in range(k):
    for j in range(i):
        if l[j] >= l[i]:
            t[i] = max(t[i],t[j]+1)
print(max(t))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
k = int(input())
l = list(map(int, input().split()))
t = [1 for _ in range(k)]
for i in range(k):
    for j in range(i):
        if l[j] >= l[i]:
            t[i] = max(t[i], t[j]+1)
print(max(t))
```

基本信息

#: 46763664

题目: 02945

提交人: 2400011474

内存: 3612kB

时间: 24ms

语言: Python3

提交时间: 2024-10-27 09:03:34

23421: 小偷背包

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

思路: dp, 物品从前往后构建二元数组, 前i个物品, j块钱, 买或不买取最大值

代码:

```
n, b = map(int, input().split())
a = list(map(int, input().split()))
c = list(map(int, input().split()))
l = [[0 for _ in range(b+1)] for _ in range(n+1)]
for i in range(1, n+1):
    for j in range(min(c[i-1], b+1)):
        l[i][j] = l[i-1][j]
    for j in range(c[i-1], b+1):
        l[i][j] = max(l[i-1][j], l[i-1][j-c[i-1]]+a[i-1])
print(l[n][b])
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
n,b = map(int,input().split())
a = list(map(int,input().split()))
c = list(map(int,input().split()))
l = [[0 for _ in range(b+1)] for _ in range(n+1)]
for i in range(1,n+1):
    for j in range(min(c[i-1],b+1)):
        l[i][j] = l[i-1][j]
    for j in range(c[i-1],b+1):
        l[i][j] = max(l[i-1][j],l[i-1][j-c[i-1]]+a[i-1])
print(l[n][b])
```

基本信息

#: 46825621
题目: 23421
提交人: 2400011474
内存: 3624kB
时间: 24ms
语言: Python3
提交时间: 2024-10-30 08:06:42

©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：递归，满足条件继续递归，到固定长度就储存，排序后输出

代码：

```
ans = []
def solve(s):
    for i in range(1,9):
        k = True
        for j in range(len(s)):
            if i == int(s[j]) or len(s)-j == abs(i-int(s[j])):
                k = False
                break
        if k:
            if len(s) == 7:
                ans.append(int(s+str(i)))
            else:
                solve(s+str(i))
solve('')
ans.sort()
n = int(input())
for _ in range(n):
    print(ans[int(input())-1])
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
ans = []
def solve(s):
    for i in range(1,9):
        k = True
        for j in range(len(s)):
            if i == int(s[j]) or len(s)-j == abs(i-int(s[j])):
                k = False
                break
        if k:
            if len(s) == 7:
                ans.append(int(s+str(i)))
            else:
                solve(s+str(i))
solve('')
ans.sort()
n = int(input())
for _ in range(n):
    print(ans[int(input())-1])
```

基本信息

#: 46829027
题目: 02754
提交人: 2400011474
内存: 3632kB
时间: 45ms
语言: Python3
提交时间: 2024-10-30 11:55:22

189A. Cut Ribbon

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

思路: dp, l是在l-x的数目里取最大的+1

代码:

```
l = list(map(int,input().split()))
n = l[0]
l = set([x for x in l[1:] if x <= n])
t = [0 for _ in range(n)]
for x in l:
    t[x-1] = 1
for i in range(n):
    for x in l:
        if i-x >= 0 and t[i-x]:
            t[i] = max(t[i],1+t[i-x])
print(t[n-1])
```

代码运行截图 (至少包含有"Accepted")

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
288651369	Practice: weijialiang	189A - 39	Python 3	Accepted	77 ms	20 KB	2024-10-29 03:16:00	2024-10-29 03:16:00		<div>Compare</div>

→ Source

Copy

```
l = list(map(int,input().split()))
n = l[0]
l = set([x for x in l[1:] if x <= n])
t = [0 for _ in range(n)]
for x in l:
    t[x-1] = 1
for i in range(n):
    for x in l:
        if i-x >= 0 and t[i-x]:
            t[i] = max(t[i],1+t[i-x])
print(t[n-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。

期中好忙，只能跟着每日选做了，算法基础就先停一段时间吧，复习烦了可以找到题做做。