

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 颜鼎堃 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:

- 递归基础练习
- 填参数的时候在草稿纸上画一画移动方法

代码:

Python

```
def Hanoi(n, stt, wth, end):
    if n == 1:
        print(f"{stt}→{end}")
    else:
        Hanoi(n-1, stt, end, wth)
        print(f"{stt}→{end}")
        Hanoi(n-1, wth, stt, end)
def main():
    n = int(input())
    print(2**n - 1)
    Hanoi(n, "A", "B", "C")
main()
```

代码运行截图（至少包含有"Accepted"）

晴问

课程 训练营 算法笔记 题库 比赛 语言入门教程 考研算法大题特训 New 本期速递

入门篇 (2) — 算法初步

递归

吓得我抱起了我的小小鲤鱼

从前有座山

序列求最大值

反转字符串

阶乘

斐波拉契数列

数塔

回文字符串

上楼

汉诺塔

棋盘覆盖问题

数字螺旋矩阵

盒分形

谢尔宾斯基地毯

自然数分解之最大积

自然数分解之方案数

递归深度

递归调试

01串

子集

题目

题解

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，在一根柱子上从下往上按照大小顺序摆着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。

输出描述

第一行输出一个整数，表示至少需要的移动次数。

接下来每行输出一行移动，格式为 $x \rightarrow y$ ，表示从柱子 x 移动最上方的圆盘到柱子 y 最上方。

样例1

代码书写

```

1 def Hanoi(n, stt, wth, end):
2     if n == 1:
3         print(f"{stt}->{end}")
4     else:
5         Hanoi(n-1, stt, end, wth)
6         print(f"{stt}->{end}")
7         Hanoi(n-1, wth, stt, end)
8 def main():
9     n = int(input())
10    print(2**n - 1)
11    Hanoi(n, "A", "B", "C")
12    main()

```

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

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

思路：

- 递归的第二个练习
- 最开始编写的时候，每一行只输出和上一行不一样的部分，连每行的长度都不一致，我选择将输出改为添加到列表，再统一修改，但是调用全局变量消耗大量时间，在 $n = 8$ 时超时
- 后来想到可以在传递参数时额外传递一项，但在每次传递前都对变量重新赋值，导致结果还是对不上，重新进行学习后发现只需改变传递时的参数表达式而不需要直接赋值，最终通过

代码：

Python

```

def permutation(n, pres = []):
    if not n:
        print(*pres, sep = " ")
    else:
        for i in n:
            permutation([ntil for ntil in n if ntil != i], pres+[i])
def main():
    permutation(list(range(1, 1+int(input()))))
main()

```

代码运行截图（至少包含有"Accepted"）

问

课程

训练营

算法笔记

题库

比赛

语言入门教程

考研算法大题特训

New 本期速递

入门篇 (2) —— 算法初步

递归

吓得我抱起了我的小小鲤鱼

从前有座山

序列求最大值

反转字符串

阶乘

斐波拉契数列

数塔

回文字符串

上楼

汉诺塔

棋盘覆盖问题

数字螺旋矩阵

盒分形

谢尔宾斯基地毯

自然数分解之最大积

自然数分解之方案数

递归深度

递归调试

01串

子集

题目

题解

全排列

通过数 1152 提交数 2397 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

输出 复制

1

代码书写

Python

```
1 def permutation(n, pres = []):
2     if not n:
3         print(*pres, sep = " ")
4     else:
5         for i in n:
6             permutation([ntil for ntil in n if ntil != i], pres + [i])
7 def main():
8     permutation(list(range(1, 1+int(input()))))
9     main()
```

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：

- 即给定一个数列，求最长单调递减子列
- 学习了一下动态规划的基本知识，这题的思路（和下题）基本都是学的OI Wiki [动态规划基础 - OI Wiki](#)

当然，本题还要感谢信科同学的帮助

- 基本思路：从后往前，数每个数后面的子列并把自己放在最长的一个的最前面
- 按照OI Wiki，复杂度为 $O(n^2)$ ，不是最佳算法，但完全能用

代码：

```
k = int(input())
mis = list(map(int, input().split()))
length = [0 for i in range(k)]
length[k-1] = 1
for i in range(k-2, -1, -1):
    maximum = 0
    for j in range(i, k):
        if mis[j] <= mis[i]:
            maximum = max(maximum, length[j])
    length[i] = maximum + 1
```

Python

```
print(max(length))
```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

 **CS101** / 计概2024fall每日选做

题目 排名 状态 提问

#46911104提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
k = int(input())
mis = list(map(int, input().split()))
length = [0 for i in range(k)]
length[k-1] = 1
for i in range(k-2, -1, -1):
    maximum = 0
    for j in range(i, k):
        if mis[j] <= mis[i]:
            maximum = max(maximum, length[j])
    length[i] = maximum + 1
print(max(length))
```

基本信息

#:

46911104

题目:

02945

提交人:

颜鼎盛(24n2400011125)

内存:

3636kB

时间:

27ms

语言:

Python3

提交时间:

2024-11-02 17:38:04

©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:

- 学习的OI Wiki
[背包 DP - OI Wiki](#)
- 转移方程: $f(j) = \max(f(j), f(j - w_i) + v_i)$, 运用二重循环

代码:

```
N, B = map(int, input().split())
p = list(map(int, input().split()))
w = list(map(int, input().split()))
maximum = [0 for j in range(B+1)]
for i in range(N):
    for j in range(B, w[i]-1, -1):
        maximum[j] = max(maximum[j], maximum[j-w[i]] + p[i])
print(max(maximum))
```

Python

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 24n2400011125 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#46924366提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
N, B = map(int, input().split())
p = list(map(int, input().split()))
w = list(map(int, input().split()))
maximum = [0 for j in range(B+1)]
for i in range(N):
    for j in range(B, w[i]-1, -1):
        maximum[j] = max(maximum[j], maximum[j-w[i]] + p[i])
print(max(maximum))
```

基本信息

#: 46924366
题目: 23421
提交人: 颜鼎壁(24n2400011125)
内存: 3620kB
时间: 24ms
语言: Python3
提交时间: 2024-11-03 13:17:41

©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:

- 感觉比全排列简单, 没借助外力自己解决了
- 先递归打表, 再直接查找 (注释部分为打表代码)
- 递归思路: n 为行数, 从 0 到 7, 将已经放好了的皇后的位置存入 `pres` 列表, 在一行中对各列进行判断, 去除不合题意的位置 (即已有的列和位于同一斜线的位置), 然后进行下一次递归, 到 $n = 8$ 时将 `pres` 作为答案, 递归结束

代码:

Python

```
queen = ['15863724', '16837425', '17468253', '17582463', '24683175', '25713864',
'25741863', '26174835', '26831475', '27368514', '27581463', '28613574', '31758246',
'35281746', '35286471', '35714286', '35841726', '36258174', '36271485', '36275184',
'36418572', '36428571', '36814752', '36815724', '36824175', '37285146', '37286415',
'38471625', '41582736', '41586372', '42586137', '42736815', '42736851', '42751863',
'42857136', '42861357', '46152837', '46827135', '46831752', '47185263', '47382516',
'47526138', '47531682', '48136275', '48157263', '48531726', '51468273', '51842736',
'51863724', '52468317', '52473861', '52617483', '52814736', '53168247', '53172864',
'53847162', '57138642', '57142863', '57248136', '57263148', '57263184', '57413862',
'58413627', '58417263', '61528374', '62713584', '62714853', '63175824', '63184275',
'63185247', '63571428', '63581427', '63724815', '63728514', '63741825', '64158273',
'64285713', '64713528', '64718253', '68241753', '71386425', '72418536', '72631485',
'73168524', '73825164', '74258136', '74286135', '75316824', '82417536', '82531746',
```

```

'83162574', '84136275']
# def eightqueens(n, pres):
#     if n == 8:
#         global queen
#         queen.append("".join(list(map(lambda t: str(t+1), pres))))
#     else:
#         for i in range(8):
#             for j in range(n):
#                 if i == pres[j] or n - i == j - pres[j] or n + i == j + pres[j]:
#                     break
#             else:
#                 eightqueens(n+1, pres+[i])

# eightqueens(0, [])
# print(queen)
for i in range(int(input())):
    print(queen[int(input())-1])

```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 24n2400011125 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#46932425提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

queen = ['15863724', '16837425', '17468253', '17582463', '24683175', '25713864',
for i in range(int(input())):
    print(queen[int(input())-1])

```

基本信息

- #: 46932425
- 题目: 02754
- 提交人: 颜鼎堃(24n2400011125)
- 内存: 3700kB
- 时间: 26ms
- 语言: Python3
- 提交时间: 2024-11-03 17:14:50

©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:

- 结合前两个学习了他人思路的题目，独立完成了本题
- 感觉似乎和完全背包问题有点像，相当于三种物品装袋子，每个物品数量无限；但事实上有着重要的区别，即袋子必须要装满（丝带最终长度归零），这导致完全背包问题的思路在此并不适用

- 事实上，本题与一道经典的小学奥数题有相似之处：一共八层楼梯，每次上一层或者两层，一共有多少种上楼方法（即斐波那契数列），但本题求的不是剪丝带方法，而是丝带段数，但都可以由初始状态递推得到
- 对应的方程：


$$f(n) = \max_{i=1,2,3,a[i] \neq 0} (f(n - a[i]) + 1, f(n))$$


代码：

Python

```
a = [0, 0, 0]
n, a[0], a[1], a[2] = map(int, input().split())
cut = [0 for i in range(n+1)]
for i in range(3):
    if a[i] <= n:
        cut[a[i]] = 1
for i in range(3):
    for j in range(n - a[i] + 1):
        if cut[j] != 0:
            cut[j + a[i]] = max(cut[j + a[i]], cut[j] + 1)
print(cut[n])
```

代码运行截图（至少包含有"Accepted"）

 **CODEFORCES**
Sponsored by TON

Chaitinien |  | [Logout](#)

HOME TOP CATALOG CONTESTS GYM **PROBLEMSET** GROUPS RATING EDU API CALENDAR HELP

MAIN

ACMSGURU

 |


PROBLEMS

SUBMIT

STATUS

STANDINGS

CUSTOM TEST

| # | Author | Problem | Lang | Verdict | Time | Memory | Sent | Judged | | |
|-----------|----------------------|---------------------------|----------|----------|-------|--------|---------------------|---------------------|---|-------------------------|
| 289783218 | Practice: Chaitinien | 189A - 39 | Python 3 | Accepted | 77 ms | 16 KB | 2024-11-03 17:15:50 | 2024-11-03 17:15:53 |  | Compare |

→ Source

Copy

```
a = [0, 0, 0]
n, a[0], a[1], a[2] = map(int, input().split())
cut = [0 for i in range(n+1)]
for i in range(3):
    if a[i] <= n:
        cut[a[i]] = 1
for i in range(3):
    for j in range(n - a[i] + 1):
        if cut[j] != 0:
            cut[j + a[i]] = max(cut[j + a[i]], cut[j] + 1)
print(cut[n])
```

[Click](#) to see test details

Codeforces

 (c) Copyright 2010-2024 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Nov/03/2024 23:49:27^{UTC+8} (12).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by





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。

1. 真心希望老师上课讲点例题，只给大方向的话，面对具体题目还是得靠自学
2. AI是人工智障我是智障人工，每天看群里怎么那么多同学反映题目简单，感觉我可以和我的期中考试成绩说再见了
3. 菜就多练