

# Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 同学的姓名、院系

## 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：这道题是很久以前做的了，今天又做了一次，有了更深的体会。当我们无法捕捉到整体的规律性时，我们可以专注于这个问题本身的自相似性化简问题，这样我们就从分析这个问题的全局转变为分析这个问题的一小部分，通过递推进而就能知道问题的全局。

代码：

```
def Hanoi(n, A, B, C):  
    if n == 1:  
        print(f'{A}->{C}')  
        return  
    Hanoi(n - 1, A, C, B)  
    print(f'{A}->{C}')  
    Hanoi(n - 1, B, A, C)  
n = int(input())  
print(2 ** n - 1)  
Hanoi(n, 'A', 'B', 'C')
```

代码运行截图 (至少包含有"Accepted")

题目

题解

汉诺塔

通过数 1395 提交数 3438 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，在一根柱子上从上往下按照大小顺序摆着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，并且任何时候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

输出描述

代码书写

```
1 def Hanoi(n, A, B, C):
2     if n == 1:
3         print(f'{A}->{C}')
4         return
5         Hanoi(n - 1, A, C, B)
6         print(f'{A}->{C}')
7         Hanoi(n - 1, B, A, C)
8     n = int(input())
9     print(2 ** n - 1)
10    Hanoi(n, 'A', 'B', 'C')
```

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

查看题解

```
used = [False] * (n + 1)
permutation(list(range(1, n + 1)), [], used)
```

代码运行截图 == (至少包含有"Accepted") ==



## 02945: 拦截导弹

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

思路：相当于最大下降子序列（注意不一定是单调的，所以会有微小的差别），有 $O(N^2)$ 和 $O(N \log N)$ 两种方法，实际差别不大。还有一种能达到 $O(N)$ 的耐心排序，没有写出来

代码：

```
import bisect
k = int(input())
a = list(map(lambda x: -int(x), input().split()))
dp = [float('inf')] * k
for i in range(k):
    dp[bisect.bisect_right(dp, a[i])] = a[i]
print(bisect.bisect_left(dp, float('inf')))
```

```
k = int(input())
a = list(map(int, input().split()))
dp = [0] * k
dp[0] = 1
for i in range(1, k):
    dp[i] = 1
    for j in range(i):
        if a[j] >= a[i]:
            dp[i] = max(dp[j] + 1, dp[i])
print(max(dp))
```

代码运行截图 (至少包含有"Accepted")



## #46828247提交状态

[查看](#) [提交](#) [统计](#) [提问](#)状态: **Accepted**

源代码

```
import bisect
k = int(input())
a = list(map(lambda x: -int(x), input().split()))
dp = [float('inf')] * k
for i in range(k):
    dp[bisect.bisect_right(dp, a[i])] = a[i]
print(bisect.bisect_left(dp, float('inf')))
```

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

基本信息

#: 46828247  
题目: 02945  
提交人: 24n2400011431  
内存: 3644kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-10-30 11:07:11

## #46828079提交状态

[查看](#) [提交](#) [统计](#) [提问](#)状态: **Accepted**

源代码

```
k = int(input())
a = list(map(int, input().split()))
dp = [0] * k
dp[0] = 1
for i in range(1, k):
    dp[i] = 1
    for j in range(i):
        if a[j] >= a[i]:
            dp[i] = max(dp[j] + 1, dp[i])
print(max(dp))
```

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

基本信息

#: 46828079  
题目: 02945  
提交人: 24n2400011431  
内存: 3648kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-10-30 11:00:52

## 23421: 小偷背包

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

思路: 经典的dp背包问题

代码:

```
n, b = map(int, input().split())
val = list(map(int, input().split()))
w = list(map(int, input().split()))
dp = [0] * (b + 1)
for i in range(n):
    for v in range(b, w[i] - 1, -1):
        dp[v] = max(dp[v - w[i]] + val[i], dp[v])
print(max(dp))
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
n, b = map(int, input().split())
val = list(map(int, input().split()))
w = list(map(int, input().split()))
dp = [0] * (b + 1)
for i in range(n):
    for v in range(b, w[i] - 1, -1):
        dp[v] = max(dp[v - w[i]] + val[i], dp[v])
print(max(dp))
```

基本信息

#: 46811776  
题目: 23421  
提交人: 24n2400011431  
内存: 3616kB  
时间: 25ms  
语言: Python3  
提交时间: 2024-10-29 14:59:13

## 02754: 八皇后

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

思路：由于答案要求得到皇后串，因此想到利用皇后串来传递皇后的位置。利用一个简单的算法来判断皇后之间是否会相互打架。然后输出即可。用皇后串而非一个表示棋盘的列表来进行dfs有一个好处：利用列表进行的话比较占内存，而且列表最终还是要转化成皇后串。

->一定要想到用皇后串进行dfs!!!

本题属于dfs中的什么类型？图的遍历/找到最短路径？->图的遍历。对什么图的遍历？对皇后串的遍历。时间复杂度为多少？比较小，因为有判断冲突以进行剪枝。

代码：

```
def dfs(s):
    for i in range(1, 9):
        flag = True
        for j in range(len(s)):
            if int(s[j]) == i or abs(len(s) - j) == abs(i - int(s[j])):
                flag = False
                break
        if flag:
            if len(s) == 7:
                ans.append(s + str(i))
            else:
                dfs(s + str(i))

ans = []
dfs('')
n = int(input())
for i in range(n):
    l = int(input())
    print(ans[l - 1])
```

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：有两种，dp或者是暴力

代码：

```
# dp做法
n, a, b, c = map(int, input().split())
dp = [float('-inf')] * (n + 1)
dp[0] = 0
for i in (a, b, c):
    for v in range(i, n + 1):
        dp[v] = max(dp[v - i] + 1, dp[v])
print(dp[-1])
```

```
# 暴力做法，可以用扩展欧几里得做法进行优化，但是这个题数据量比较小，没什么必要。
n, a, b, c = map(int, input().split())
a, b, c = sorted([a, b, c])
ans = 0
for i in range(n // a + 1):
    for j in range((n - i * a) // b, -1, -1):
        if (n - i * a - j * b) % c == 0:
            ans = max(ans, i + j + (n - i * a - j * b) // c)
            break
print(ans)
```

代码运行截图 (至少包含有"Accepted")

## 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。

善用AI辅助做题，对自己的代码进行优化和改进。

**学习计概之路道阻且长，在学习和做题过程中总会遇到不顺，总会有崩溃的瞬间。但是当自己想出一个能正确解决题目的算法时获得的成就感会让这一切值得。与诸君共勉！**