

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by <mark>祁黄奕 城市与环境学院</mark>

****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

syl119: 汉诺塔

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

思路：实际上每个问题都可以表述为：将 $n-1$ 个盘子移到中间柱子；将 i 个盘子移到目标柱子；再更换中间柱子与起始柱子，重复该过程。

代码：

```
```python
def Hanoi(x,a,b,c):
 if x==1:
 print(f"{a}->{c}")
 return
 Hanoi(x-1,a,c,b)
```

```

 print(f"{a}->{c}")
 Hanoi(x-1,b,a,c)
n=int(input())
print(pow(2,n)-1)
Hanoi(n,"A","B","C")
'''

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

代码书写

Python

```

1 def Hanoi(x,a,b,c):
2 if x==1:
3 print(f"{a}->{c}")
4 return
5 Hanoi(x-1,a,c,b)
6 print(f"{a}->{c}")
7 Hanoi(x-1,b,a,c)
8 n=int(input())
9 print(pow(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

查看题解

100% 数据通过测试

运行时长: 0 ms

### sy132: 全排列 I

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

思路：每次排第  $i$  位，然后再去排第  $i+1$  位，以此类推，但排第  $i+1$  位时不能与前  $i$  位重复，所以存在标记和回溯

代码：

```
```python
```

```
def arrange(numlst,path):  
    if len(path)==len(numlst):  
        print(*path)  
        return  
    for i in range(len(numlst)):  
        if numlst[i]:  
            para=numlst[i]  
            numlst[i]=False  
            path.append(para)  
            arrange(numlst,path)  
            numlst[i]=para  
            path.pop()  
n=int(input())  
nums=[i for i in range (1,n+1)]  
arrange(nums, [])  
```
```

代码运行截图 ==（至少包含有“Accepted”）==

代码书写

Python

```
1 def arrange(numlst,path):
2 if len(path)==len(numlst):
3 print(*path)
4 return
5 for i in range(len(numlst)):
6 if numlst[i]:
7 para=numlst[i]
8 numlst[i]=False
9 path.append(para)
10 arrange(numlst,path)
11 numlst[i]=para
12 path.pop()
13 n=int(input())
14 nums=[i for i in range (1,n+1)]
15 arrange(nums, [])
```

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

### ### 02945: 拦截导弹

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

思路：找出最长的非递减子序列长度。

代码：

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

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#### #46923850提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

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

基本信息

#: 46923850  
题目: 02945  
提交人: qhy  
内存: 3628kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-11-03 12:39:02

©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：建立二维列表，

代码：

```
```python
```

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

```
```
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#### #46923986提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

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

基本信息

#: 46923986  
题目: 23421  
提交人: qhy  
内存: 3560kB  
时间: 25ms  
语言: Python3  
提交时间: 2024-11-03 12:49:08

©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：使用深度优先搜索（DFS）生成所有可能的 8 皇后解。尝试在当前行的每一列放置皇后。初始化标志变量 flag 为 True，表示当前列可以放置皇后。检查当前列是否与已放置的皇后冲突、是否有其他皇后、是否在同一对角线上。如果有冲突，将 flag 设为 False 并跳出循环。没有冲突则继续处理。如果当前解已经包含了 7 个皇后，添加最后一个皇后并将其加入解列表 ans。否则，递归调用 dfs 继续放置下一个皇后。

代码：

```
```python
```

```
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 <mark>（至少包含有“Accepted”）</mark>

#46924150提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

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])
'''

```

基本信息

#: 46924150
 题目: 02754
 提交人: qhy
 内存: 3620kB
 时间: 44ms
 语言: Python3
 提交时间: 2024-11-03 13:03:28

©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：dp 外层循环遍历每个可能的分割长度 i （即 a 、 b 、 c ）。内层循环遍历从 i 到 n 的所有长度 v 。对于每个长度 v ，更新 $dp[v]$ 为 $\max(dp[v - i] + 1, dp[v])$ ： $dp[v - i] + 1$ 表示在长度 $v - i$ 的基础上再加一段长度为 i 的段，分割段数加 1。 $dp[v]$ 表示当前长度 v 的最大分割段数。取这两者的最大值作为新的 $dp[v]$ 。

代码：

```

python
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])

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

MAIN

ACHMSGURU

PROBLEMS

SUBMIT

STATUS

STANDINGS

CUSTOM TEST

☒ my only
 ☐ Friends only

Contest status

#	When	Who	Problem	Lang	Verdict	Time	Memory
289720429	Nov/03/2024 13:18 UTC+8	qhy06	189A - Cut Ribbon	Python 3	Accepted	62 ms	0 KB

2. 学习总结和收获

dp、递归、深度搜索都好难好难好难，接近期中忙到根本没时间继续每日选做，看题解做题也依然很难理解（或许再多看点题解、多做点题能让我更了解吧）。稍微对 dp 的运作方法有了一点点认识，最多看到题之后有大概思路，但自己写不明白导致最后还是需要看大佬的题解和跑 ai……估计考试看到类似这次作业的题就直接放弃了……