

# Assignment #3: March月考

---

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by 武昱达 23工院

## 说明:

- 1) The complete process to learn DSA from scratch can be broken into 4 parts:
  - Learn about Time and Space complexities
  - Learn the basics of individual Data Structures
  - Learn the basics of Algorithms
  - Practice Problems on DSA
- 2) 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。

## 编程环境

Python编程环境：PyCharm 2023.3.4 (Professional Edition) & VSCode

操作系统：Windows 11

## 1. 题目

---

### 02945: 拦截导弹

<http://cs101.openjudge.cn/practice/02945/>

思路：

之前写过dp实现，由于数据量较小，这次使用dfs实现。

### 代码

```
1 # dfs
2 import math
3 k=int(input())
4 heights=list(map(int,input().split()))
5 res=0
6 def dfs(path,left):
7     global res
8     if len(left)==0:
9         res=max(res,len(path))
10        return
11    if len(left)==1:
```

```

12         height=left.pop()
13         if height<path[-1]:
14             path.append(height)
15         res=max(len(path),res)
16         return
17     for i in range(len(left)):
18         height=left[i]
19         if height<=path[-1] or len(path)==0:
20             dfs(path+[height],left[i+1:])
21     return
22
23 dfs([math.inf],heights)
24 print(res-1)

```

代码运行截图 (至少包含有"Accepted")

#44090336提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

import math
# dfs实现
k=int(input())
heights=list(map(int,input().split()))
res=0
def dfs(path,left):
    global res
    if len(left)==0:
        res=max(res,len(path))
        return
    if len(left)==1:
        height=left.pop()
        if height<path[-1]:
            path.append(height)
            res=max(len(path),res)
            return
    for i in range(len(left)):
        height=left[i]
        if height<=path[-1] or len(path)==0:
            dfs(path+[height],left[i+1:])
    return

dfs([math.inf],heights)
print(res-1)

```

基本信息

#: 44090336  
 题目: E02945  
 提交人: 23n2300011119 (武)  
 内存: 3604kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2024-03-06 15:37:39

04147:汉诺塔问题(Tower of Hanoi)

<http://cs101.openjudge.cn/practice/04147>

思路:

递归。

## 代码

```
1 # 通过不断变化initiate, assistance和goal实现递归。
2 def OneMove(x,init,goal):
3     print(str(x)+":"+init+"->" +goal)
4 def Move(num_disks,init,assist,goal):
5     if num_disks == 1:
6         OneMove(num_disks,init,goal)
7     else:
8         Move(num_disks-1,init,goal,assist)
9         OneMove(num_disks,init,goal)
10        Move(num_disks-1,assist,init,goal)
11
12 n,a,b,c=map(str,input().split())
13 Move(int(n),a,b,c)
```

代码运行截图 (至少包含有"Accepted")

### #44093350提交状态

查看 提交 统计 提问

状态: **Accepted**

源代码

```
def OneMove(x,init,goal):
    print(str(x)+":"+init+"->" +goal)
def Move(num_disks,init,assist,goal):
    if num_disks == 1:
        OneMove(num_disks,init,goal)
    else:
        Move(num_disks-1,init,goal,assist)
        OneMove(num_disks,init,goal)
        Move(num_disks-1,assist,init,goal)

n,a,b,c=map(str,input().split())
Move(int(n),a,b,c)
```

基本信息

#: 44093350  
题目: 04147  
提交人: 23n2300011119 (武)  
内存: 3536kB  
时间: 20ms  
语言: Python3  
提交时间: 2024-03-06 17:12:43

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## 03253: 约瑟夫问题No.2

<http://cs101.openjudge.cn/practice/03253>

思路:

简单的队列实现。

## 代码

```
1 # queue
2 from collections import deque
3 while True:
4     n,p,m=map(int,input().split())
5     res=[]
6     if n==0:break
7     q=deque(i for i in range(1,n+1))
8     for _ in range(p - 1):
9         tmp = q.popleft()
10        q.append(tmp)
11    while len(q)>0:
12        for _ in range(m-1):
13            tmp=q.popleft()
```

```
14         q.append(tmp)
15         res.append(str(q.popleft()))
16     print("".join(res))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

### #44090079提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
while True:
    n,p,m=map(int,input().split())
    res=[]
    if n==0:break
    q=deque(i for i in range(1,n+1))
    for _ in range(p - 1):
        tmp = q.popleft()
        q.append(tmp)
    while len(q)>0:
        for _ in range(m-1):
            tmp=q.popleft()
            q.append(tmp)
        res.append(str(q.popleft()))
    print("".join(res))
```

基本信息

#: 44090079

题目: M03253

提交人: 23n2300011119 (武)

内存: 3608kB

时间: 20ms

语言: Python3

提交时间: 2024-03-06 15:27:31

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## 21554:排队做实验 (greedy)v0.2

<http://cs101.openjudge.cn/practice/21554>

思路:

很简单的贪心, 用反证法可以轻松证明。

代码

```
1  # greedy
2  n,res,wait=int(input()),[],0
3  time=[0]+list(map(int,input().split()))
4  t_time=list((time[i],i) for i in range(1,n+1))
5  t_time.sort()
6  # print(t_time)
7  for i in range(n):
8      res.append(t_time[i][1])
9      wait+=t_time[i][0]*(n-1-i)
10 print(*res)
11 print('{:.2f}'.format(wait/n))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
n,res,wait=int(input()),[],0
time=[0]+list(map(int,input().split()))
t_time=list((time[i],i) for i in range(1,n+1))
t_time.sort()
# print(t_time)
for i in range(n):
    res.append(t_time[i][1])
    wait+=t_time[i][0]*(n-1-i)
print(*res)
print('{:.2f}'.format(wait/n))
```

基本信息

#: 44090667  
题目: M21554  
提交人: 23n2300011119 (武)  
内存: 3616kB  
时间: 20ms  
语言: Python3  
提交时间: 2024-03-06 15:52:17

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 19963:买学区房

<http://cs101.openjudge.cn/practice/19963>

思路:

语法题，但是很烦。

代码

```
1 def mid_num(lst):
2     tmp=sorted(lst)
3     if len(lst)%2==0:
4         return (tmp[len(lst)//2]+tmp[len(lst)//2-1])/2
5     if len(lst)%2==1:
6         return tmp[len(lst)//2]
7
8 n,res=int(input()),0
9 d_raw=list(map(str,input().split()))
10 d_1=[]
11
12 for i in range(n):
13     a,b=map(str,d_raw[i].split(","))
14     a,b=int(a[1:]),int(b[:len(b)-1])
15     d_1.append((a,b))
16
17 p=list(map(int,input().split()))
18 d=[i+j for i,j in d_1]
19 p_mid=mid_num(p)
20 x=[]
21 for i in range(n):
22     xt=d[i]/p[i]
23     x.append(xt)
24 x_mid=mid_num(x)
25
26 for i,j in zip(x,p):
27     if i>x_mid and j<p_mid:
28         res+=1
29 print(res)
```

## #44091503提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def mid_num(lst):
    tmp=sorted(lst)
    if len(lst)%2==0:
        return (tmp[len(lst)//2]+tmp[len(lst)//2-1])/2
    if len(lst)%2==1:
        return tmp[len(lst)//2]

n,res=int(input()),0
d_raw=list(map(str,input().split()))
d_1=[]

for i in range(n):
    a,b=map(str,d_raw[i].split(", "))
    a,b=int(a[1:]),int(b[:len(b)-1])
    d_1.append((a,b))

p=list(map(int,input().split()))
d=[i+j for i,j in d_1]
p_mid=mid_num(p)
x=[]
for i in range(n):
    xt=d[i]/p[i]
    x.append(xt)
x_mid=mid_num(x)

for i,j in zip(x,p):
    if i>x_mid and j<p_mid:
        res+=1
print(res)
```

基本信息

#: 44091503  
 题目: T19963  
 提交人: 23n2300011119 (武)  
 内存: 4432kB  
 时间: 26ms  
 语言: Python3  
 提交时间: 2024-03-06 16:21:17

## 27300: 模型整理

<http://cs101.openjudge.cn/practice/27300>

思路:

写一个排序函数, 结合字典即可, defaultdict好用。

代码

```
1 def sort_str(scales):
2     M_part,B_part=[],[]
3     for i in scales:
4         if i[-1]=="M":
5             M_part.append(i)
6         elif i[-1]=="B":
7             B_part.append(i)
8     B_part.sort(key=lambda x:float(x[:len(x)-1]))
9     M_part.sort(key=lambda x:float(x[:len(x)-1]))
10    return M_part+B_part
11
12 from collections import defaultdict
13 models=defaultdict(list)
14 for _ in range(n:=int(input())):
15     name,scale=map(str,input().split("-"))
16     models[name].append(scale)
17
18 lst=[]
19 for name,scales in models.items():
20     lst.append((name,scales))
21 lst.sort(key=lambda x:x[0])
```

```
22 for name,scales in lst:
23     print(name+": "+", ".join(sort_str(scales)))
```

代码运行截图 (AC代码截图，至少包含有"Accepted")

#44091843提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def sort_str(scales):
    M_part,B_part=[],[]
    for i in scales:
        if i[-1]=="M":
            M_part.append(i)
        elif i[-1]=="B":
            B_part.append(i)
    B_part.sort(key=lambda x:float(x[:len(x)-1]))
    M_part.sort(key=lambda x:float(x[:len(x)-1]))
    return M_part+B_part

from collections import defaultdict
models=defaultdict(list)
for _ in range(n:=int(input())):
    name,scale=map(str,input().split("-"))
    models[name].append(scale)

lst=[]
for name,scales in models.items():
    lst.append((name,scales))
lst.sort(key=lambda x:x[0])
for name,scales in lst:
    print(name+": "+", ".join(sort_str(scales)))
```

基本信息

#: 44091843

题目: T27300

提交人: 23n2300011119 (武)

内存: 3672kB

时间: 23ms

语言: Python3

提交时间: 2024-03-06 16:32:13

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## 2. 学习总结和收获

一直在跟进每日选做，目前要求自己每个题目都完成。

最近题目难度上来，不如前几天游刃有余，也是因为上学期每日选做做的很少（欠的债总是要还）。争取持续跟进。

目前仍然很难克服对递归的恐惧（同一函数体容易想不清楚、写不清楚），即使是简单的dfs不参考模版也写得很慢，仍然需要加强练习。汉诺塔问题没有写出来遗憾AC5。

快速堆猪学会了懒删除堆，非常好用。