

Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by 同学的姓名、院系

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

(请改为同学的操作系统、编程环境等)

操作系统: macOS Ventura 13.4.1 (c)

Python编程环境: Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境: Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

02945: 拦截导弹

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

思路: 感觉昏昏的不是很会, 我先得把递归, 深度搜索仔细复习一遍。

代码

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

代码运行截图 (至少包含有"Accepted")

#44165858提交状态

状态: Accepted

源代码

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

04147:汉诺塔问题(Tower of Hanoi)

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

思路: 同第一题

代码

```
# def moveOne(numDisk: int, init: str, desti: str):
#     print("{}: {}->{}".format(numDisk, init, desti))

def move(numDisks: int, init: str, temp: str, desti: str):
    if numDisks == 1:
        moveOne(1, init, desti)
    else:
```

```
        move(numDisks - 1, init, desti, temp)

        moveOne(numDisks, init, desti)

        move(numDisks - 1, temp, init, desti)

n, a, b, c = input().split()
move(int(n), a, b, c)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
def moveOne(numDisk: int, init: str, desti: str):
    print("{}: {}->{}".format(numDisk, init, desti))

def move(numDisks: int, init: str, temp: str, desti: str):
    if numDisks == 1:
        moveOne(1, init, desti)
    else:
        move(numDisks - 1, init, desti, temp)

        moveOne(numDisks, init, desti)

        move(numDisks - 1, temp, init, desti)

n, a, b, c = input().split()
move(int(n), a, b, c)
```

2002-2022 POJ 京ICP备20010980号-1

03253: 约瑟夫问题No.2

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

思路: 利用双端队列可以实现, 只不过代码部分不是很好实现, 想明白了, 但是自己没写明白。

代码

```
# while True:
    n, p, m = map(int, input().split())
    if {n,p,m} == {0}:
        break
    monkey = [i for i in range(1, n+1)]
    for _ in range(p-1):
        tmp = monkey.pop(0)
        monkey.append(tmp)
    # print(monkey)

    index = 0
    ans = []
    while len(monkey) != 1:
        temp = monkey.pop(0)
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        monkey.append(temp)

    ans.extend(monkey)

    print(' '.join(map(str, ans)))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
while True:
    n, p, m = map(int, input().split())
    if {n,p,m} == {0}:
        break
    monkey = [i for i in range(1, n+1)]
    for _ in range(p-1):
        tmp = monkey.pop(0)
        monkey.append(tmp)
        # print(monkey)

    index = 0
    ans = []
    while len(monkey) != 1:
        temp = monkey.pop(0)
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        monkey.append(temp)

    ans.extend(monkey)

    print(', '.join(map(str, ans)))
```

©2002-2022 POJ 京ICP备20010980号-1

21554:排队做实验 (greedy)v0.2

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

思路: 贪心算法, 先让时间短的做

代码

```
#
# 贪心算法可以试试
n = int(input())
*L, = map(int,input().split())
od = sorted(range(1, n+1), key = lambda x: L[x - 1])
L.sort()
t = sum((n-i-1) * L[i] for i in range(n)) / n

print(*od)
print('{:.2f}'.format(t))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
# 贪心算法可以试试
n = int(input())
*L, = map(int, input().split())
od = sorted(range(1, n+1), key = lambda x: L[x - 1])
L.sort()
t = sum((n-i-1) * L[i] for i in range(n)) / n

print(*od)
print('{:.2f}'.format(t))
```

19963:买学区房

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

思路: 按照题意求中位数, 题目比较难以阅读, 但是思路较为简单。

代码

```
# n = int(input())
dis = [eval(x)[0] + eval(x)[1] for x in input().split()]
pri = [int(x) for x in input().split()]
vau = [dis[x] / pri[x] for x in range(n)]

def mid(n, lis):
    lis = sorted(lis)
    if n % 2 == 1:
        return lis[n // 2]
    else:
        return (lis[n // 2 - 1] + lis[n // 2]) / 2

prim = mid(n, pri)
vaum = mid(n, vau)
sum = 0
for i in range(n):
    if vau[i] > vaum and pri[i] < prim:
        sum += 1
print(sum)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
dis = [eval(x)[0] + eval(x)[1] for x in input().split()]
pri = [int(x) for x in input().split()]
vau = [dis[x] / pri[x] for x in range(n)]

def mid(n, lis):
    lis = sorted(lis)
    if n % 2 == 1:
        return lis[n // 2]
    else:
        return (lis[n // 2 - 1] + lis[n // 2]) / 2

prim = mid(n, pri)
vaum = mid(n, vau)
sum = 0
for i in range(n):
    if vau[i] > vaum and pri[i] < prim:
        sum += 1
print(sum)
```

©2002-2022 POJ 京ICP备20010980号-1

27300: 模型整理

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

思路: 当时来不及做了, 但是现在看来比较简单。

代码

```
#
from collections import defaultdict

n = int(input())
d = defaultdict(list)
for _ in range(n):
    # name, para = input().strip().split('-')
    name, para = input().split('-')
    if para[-1] == 'M':
        d[name].append((para, float(para[:-1]) / 1000))
    else:
        d[name].append((para, float(para[:-1])))

sd = sorted(d)
# print(d)
for k in sd:
    paras = sorted(d[k], key=lambda x: x[1])
```

```
# print(paras)
value = ', '.join([i[0] for i in paras])
print(f'{k}: {value}')
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import defaultdict

n = int(input())
d = defaultdict(list)
for _ in range(n):
    #name, para = input().strip().split('-')
    name, para = input().split('-')
    if para[-1] == 'M':
        d[name].append((para, float(para[:-1])/1000) )
    else:
        d[name].append((para, float(para[:-1])))

sd = sorted(d)
#print(d)
for k in sd:
    paras = sorted(d[k], key=lambda x: x[1])
    #print(paras)
    value = ', '.join([i[0] for i in paras])
    print(f'{k}: {value}')
```

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“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

本周学习了线性数据结构的一些基础应用, 复习了递归和dfs的一些题目, 因为之前没有怎么接触过这类题目, 所以对我来说还是比较陌生。加上这周末感冒拉肚子耽误半天, 导致思考时间有限。实在抱歉, 没有完成自己的目标。而对于第一个导弹问题我现在都不是很懂其中原理, 打算今天再学2个小时的dp和dfs。闫先生如果有基础一点的讲义或者题目将感激不尽。