

Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by ==陈奕好==

==AC 6==

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 Sonoma 14.3.1 (c)

Python编程环境：PyCharm 2023.3.1 (Professional Edition)

1. 题目

02945: 拦截导弹

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

思路：最长下降子序列

代码

```
N = int(input())
nums = list(map(int, input().split())) # 输入一组序列
length = len(nums)
# print(n)

dp = [1] * (length + 1)

for i in range(length):
    for j in range(0, i):
        if nums[i] <= nums[j]:
            # 状态: dp[i] 表示以 nums[i] 结尾的「上升子序列」的长度
            # 当nums[i]前面存在小于nums[i]的nums[j],
            # 则暂存在dp[j]+1就是当前nums[i]的最长增长子序列的长度
            dp[i] = max(dp[i], dp[j] + 1)

print(max(dp)) # 用函数max直接找到dp数组的最大值, 无需再遍历了
```

代码运行截图 == (至少包含有"Accepted") ==

#44093734提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
N = int(input())
nums = list(map(int, input().split())) # 输入一组序列
length = len(nums)
# print(n)

dp = [1] * (length + 1)

for i in range(length):
    for j in range(0, i):
        if nums[i] <= nums[j]:
            # 状态: dp[i] 表示以 nums[i] 结尾的「上升子序列」的长度
            # 当nums[i]前面存在小于nums[i]的nums[j],
            # 则暂存在dp[j]+1就是当前nums[i]的最长增长子序列的长度
            dp[i] = max(dp[i], dp[j] + 1)

print(max(dp)) # 用函数max直接找到dp数组的最大值, 无需再遍历了
```

基本信息

#: 44093734
题目: 02945
提交人: 23n2300011030(陈奕好)
内存: 3564kB
时间: 22ms
语言: Python3
提交时间: 2024-03-06 17:24:37

04147:汉诺塔问题(Tower of Hanoi)

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

思路: 很难实现的递归

代码

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

def moveOne(x, init, desti):
    print(f"{x}:{init}->{desti}")
    return

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

move(numDisks, a, b, c)
```

代码运行截图 == (至少包含有"Accepted") ==

#44093763提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

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

def moveOne(x, init, desti):
    print(f"{x}:{init}->{desti}")
    return

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

move(numDisks, a, b, c)
```

基本信息

#: 44093763
题目: 04147
提交人: 23n2300011030(陈奕好)
内存: 3580kB
时间: 22ms
语言: Python3
提交时间: 2024-03-06 17:25:30

03253: 约瑟夫问题No.2

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

思路：加一个开始

代码

```
from collections import deque
while True:
    n, q, m = map(int, input().split())
    if {n,q,m} == {0}:
        break
    monkey = deque(i for i in range(1, n+1))
    for i in range(q-1):
        monkey.append(monkey.popleft())
    index = 0
    ans = []
    while len(monkey) != 1:
        temp = monkey.popleft()
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        monkey.append(temp)
    ans.append(monkey[0])
    print(",".join(str(_) for _ in ans))
```

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 deque
while True:
    n, q, m = map(int, input().split())
    if {n,q,m} == {0}:
        break
    monkey = deque(i for i in range(1, n+1))
    for i in range(q-1):
        monkey.append(monkey.popleft())
    index = 0
    ans = []
    while len(monkey) != 1:
        temp = monkey.popleft()
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        monkey.append(temp)
    ans.append(monkey[0])
    print(" ".join(str(_) for _ in ans))
```

基本信息

#: 42917109
题目: 03253
提交人: 23n2300011030(陈奕好)
内存: 3612kB
时间: 23ms
语言: Python3
提交时间: 2023-12-03 16:44:24

21554:排队做实验 (greedy)v0.2

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

思路：排序是最好的办法

代码

```
n = int(input())
task_time = list(map(int, input().split()))
for i in range(n):
    task_time[i] = (task_time[i], i + 1)

def minimumWaitingTime(queries):
    ans = []
    queries = sorted(queries)
    total_waiting_time = 0
    for index, time_cost_index in enumerate(queries):
        # print(f"index: {index} time_cost: {time_cost}")
        queries_Left = len(queries)-(index+1)
        total_waiting_time += time_cost_index[0] * queries_Left
        ans.append(time_cost_index[1])
    print(*ans)
    print("%.2f" % (total_waiting_time/n))
```

```
minimumWaitingTime(task_time)
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#44093819提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

基本信息

#: 44093819

题目: 21554

提交人: 23n2300011030(陈奕好)

内存: 3632kB

时间: 23ms

语言: Python3

提交时间: 2024-03-06 17:28:02

```
n = int(input())
task_time = list(map(int, input().split()))
for i in range(n):
    task_time[i] = (task_time[i], i + 1)

def minimumWaitingTime(queries):
    ans = []
    queries = sorted(queries)
    total_waiting_time = 0
    for index, time_cost_index in enumerate(queries):
        # print(f"index: {index} time_cost: {time_cost}")
        queries_Left = len(queries) - (index + 1)
        total_waiting_time += time_cost_index[0] * queries_Left
        ans.append(time_cost_index[1])
    print(*ans)
    print("%.2f" % (total_waiting_time / n))

minimumWaitingTime(task_time)
```

©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: 要存的太多了

代码

```
n = int(input())
x_y = [tuple(map(int, i[1:-1].split(","))) for i in input().split()]
# print(type(x_y))
cost = list(map(int, input().split()))
sorted_cost = sorted(cost)
ptov = []

house = []
for i in range(n):
    x, y = x_y[i][0], x_y[i][1]
    p2v = (x+y)/cost[i]
    house.append((p2v, cost[i]))
```

```

ptov.append(p2v)

sorted_ptov = sorted(ptov)

if n % 2 == 0:
    mid_cost = sorted_cost[len(sorted_cost)//2] + sorted_cost[len(sorted_cost)//2 - 1]
    mid_cost /= 2
    mid_p2v = sorted_ptov[len(sorted_ptov)//2] + sorted_ptov[len(sorted_ptov)//2 - 1]
    mid_p2v /= 2
else:
    mid_cost = sorted_cost[len(sorted_cost)//2]
    mid_p2v = sorted_ptov[len(sorted_ptov)//2]
ans = 0
for i in house:
    if mid_cost > i[1] and mid_p2v < i[0]:
        ans += 1
print(ans)
# print(x_y)

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#44093851提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

基本信息

源代码

#: 44093851

题目: 19963

提交人: 23n2300011030(陈奕好)

内存: 4432kB

时间: 25ms

语言: Python3

提交时间: 2024-03-06 17:29:34

```

n = int(input())
x_y = [tuple(map(int, i[1:-1].split(","))) for i in input().split()]
# print(type(x_y))
cost = list(map(int, input().split()))
sorted_cost = sorted(cost)
ptov = []

house = []
for i in range(n):
    x, y = x_y[i][0], x_y[i][1]
    p2v = (x+y)/cost[i]
    house.append((p2v, cost[i]))
    ptov.append(p2v)

sorted_ptov = sorted(ptov)

if n % 2 == 0:
    mid_cost = sorted_cost[len(sorted_cost)//2] + sorted_cost[len(sorted_cost)//2 - 1]
    mid_cost /= 2
    mid_p2v = sorted_ptov[len(sorted_ptov)//2] + sorted_ptov[len(sorted_ptov)//2 - 1]
    mid_p2v /= 2
else:
    mid_cost = sorted_cost[len(sorted_cost)//2]
    mid_p2v = sorted_ptov[len(sorted_ptov)//2]
ans = 0
for i in house:
    if mid_cost > i[1] and mid_p2v < i[0]:
        ans += 1
print(ans)
# print(x_y)

```

27300: 模型整理

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

思路：字符串处理

代码

```
n = int(input())
dic = dict()
for i in range(n):
    name, params = input().split("-")
    x = float(params[:-1])
    x = int(x) if x.is_integer() else float(x)
    if name in dic.keys():
        if params[-1] == "M":
            dic[name][0].append(x)
        else:
            dic[name][1].append(x)
    else:
        if params[-1] == "M":
            dic[name] = [[x], []]
        else:
            dic[name] = [[], [x]]

ans_l = []
for name, params in dic.items():
    params[0].sort()
    params[1].sort()
    # print(params)
    ans = f"{name}: "
    tmp = []
    for m in params[0]:
        tmp.append(str(m) + "M")
    for b in params[1]:
        tmp.append(str(b) + "B")
    ans += ", ".join(tmp)
    ans_l.append(ans)
ans_l.sort()
for i in ans_l:
    print(i)
```

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())
dic = dict()
for i in range(n):
    name, params = input().split("-")
    x = float(params[:-1])
    x = int(x) if x.is_integer() else float(x)
    if name in dic.keys():
        if params[-1] == "M":
            dic[name][0].append(x)
        else:
            dic[name][1].append(x)
    else:
        if params[-1] == "M":
            dic[name] = [[x], []]
        else:
            dic[name] = [[], [x]]

ans_l = []
for name, params in dic.items():
    params[0].sort()
    params[1].sort()
    # print(params)
    ans = f"{name}: "
    tmp = []
    for m in params[0]:
        tmp.append(str(m) + "M")
    for b in params[1]:
        tmp.append(str(b) + "B")
    ans += ", ".join(tmp)
    ans_l.append(ans)
ans_l.sort()
for i in ans_l:
```

基本信息

#: 44093880
题目: 27300
提交人: 23n2300011030(陈奕好)
内存: 3680kB
时间: 22ms
语言: Python3
提交时间: 2024-03-06 17:31:01

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。
==

考试有点难度，主要是汉诺塔有点记不起来了。

排队想了一会，没想到猜对了。

选做还在坚持，感觉专业课多起来后有点忙了。（逃