

Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by 李佳霖, 心理与认知科学学院

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

(请改为同学的操作系统、编程环境等)

操作系统: macOS Sonoma 14.3

Python编程环境: VSCode

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：类似于最长下降子序列问题，采用dp的思想，通过列表记录第j个元素时的最长不增子序列，在判断新的元素满足条件后，则直接继承之前计算过的序列长度，从而减少时间复杂度。

代码

```
#
k = int(input())
m = list(map(int, input().split()))
dp = [1]*k # 用于记录截止到某一位置时的最大不增子序列
for i in range(k):
    for j in range(i):
        if m[j]>=m[i]: #为了确认列表中第i个元素是否比第j个元素小，若小，则直接继承之前计算过的到第j个位置
            的最长不增子序列
            dp[i] = max(dp[j]+1, dp[i]) # dp[j]+1便为到第j个元素的最长不增子序列加上第i个元素后的新构
            成的子序列
res = max(dp)
print(res)
```

代码运行截图 (至少包含有"Accepted")

#44130374提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
k = int(input())
m = list(map(int, input().split()))
dp = [1]*k # 用于记录截止到某一位置时的最大不增子序列
for i in range(k):
    for j in range(i):
        if m[j]>=m[i]: #为了确认列表中第i个元素是否比第j个元素小，若小，则直接继
            dp[i] = max(dp[j]+1, dp[i]) # dp[j]+1便为到第j个元素的最长不增子
res = max(dp)
print(res)
```

基本信息

#: 44130374
题目: 02945
提交人: 李佳霖2000013713
内存: 3616kB
时间: 20ms
语言: Python3
提交时间: 2024-03-09 13:26:26

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

04147:汉诺塔问题(Tower of Hanoi)

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

思路: 经典的递归问题

代码

```
#
numDisks, init, temp, desti = input().split()
numDisks = int(numDisks)

def moveOne(numDisk, strinit, strdesti):
    print('{}: {}->{}'.format(numDisk, strinit, strdesti))

def move(numDisks, strinit, strtemp, strdesti):
    if numDisks == 1:
        moveOne(numDisks, strinit, strdesti)
    else:
        move(numDisks-1, strinit, strdesti, strtemp) # 把前面n-1个盘子借助第3座移动到第2座
        moveOne(numDisks, strinit, strdesti) # 把第1座上的第n个盘子移动到第3座
        move(numDisks-1, strtemp, strinit, strdesti) # 把第2座上的n-1个盘子借助第1座移动到第3座

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

代码运行截图 (至少包含有"Accepted")

#44131142提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
numDisks, init, temp, desti = input().split()
numDisks = int(numDisks)

def moveOne(numDisk, strinit, strdesti):
    print('{}: {}->{}'.format(numDisk, strinit, strdesti))

def move(numDisks, strinit, strtemp, strdesti):
    if numDisks == 1:
        moveOne(numDisks, strinit, strdesti)
    else:
        move(numDisks-1, strinit, strdesti, strtemp) # 把前面n-1个盘子借助
        moveOne(numDisks, strinit, strdesti) # 把第1座上的第n个盘子移动到第3
        move(numDisks-1, strtemp, strinit, strdesti) # 把第2座上的n-1个盘

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

基本信息

#: 44131142
题目: 04147
提交人: 李佳霖2000013713
内存: 3620kB
时间: 22ms
语言: Python3
提交时间: 2024-03-09 14:01:39

©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: 偷懒直接用了标准库collections中的deque队列结构

代码

```
#
from collections import deque
while True:
    n, p, m = map(int, input().split())
    if n == 0 and p == 0 and m == 0:
        break
    childs = deque([_ for _ in range(1,n+1)])
    res = []

    for i in range(1,p):
        child = childs[0]
        childs.popleft()
        childs.append(child)

    while len(childs)>0:
        for i in range(1,m):
            child =childs[0]
            childs.popleft()
            childs.append(child)
        leftchild = childs[0]
        childs.popleft()
        res.append(str(leftchild))

    print(','.join(res))
```

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, p, m = map(int, input().split())
    if n == 0 and p == 0 and m == 0:
        break
    childs = deque([_ for _ in range(1, n+1)])
    res = []

    for i in range(1, p):
        child = childs[0]
        childs.popleft()
        childs.append(child)

    while len(childs) > 0:
        for i in range(1, m):
            child = childs[0]
            childs.popleft()
            childs.append(child)
        leftchild = childs[0]
        childs.popleft()
        res.append(str(leftchild))

    print(','.join(res))
```

基本信息

#: 44131985
题目: 03253
提交人: 李佳霖2000013713
内存: 3980kB
时间: 22ms
语言: Python3
提交时间: 2024-03-09 14:31:03

21554:排队做实验 (greedy)v0.2

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

思路：将列表从小到大排序即可

代码

```
#
n=int(input())
s=[int(x) for x in input().split()]
ss,q=[],[]
ans=0
for i in range(n):
    ss.append([s[i],i+1])
ss.sort()
for i in range(n):
    q.append(ss[i][1])
    ans+=(n-i-1)*ss[i][0]
print(' '.join(str(x) for x in q))
print("{:.2f}".format(ans/n))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

#44132265提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n=int(input())
s=[int(x) for x in input().split()]
ss,q=[],[]
ans=0
for i in range(n):
    ss.append([s[i],i+1])
ss.sort()
for i in range(n):
    q.append(ss[i][1])
    ans+=(n-i-1)*ss[i][0]
print(' '.join(str(x) for x in q))
print("{:.2f}".format(ans/n))
```

基本信息

#: 44132265
 题目: 21554
 提交人: 李佳霖2000013713
 内存: 3624kB
 时间: 21ms
 语言: Python3
 提交时间: 2024-03-09 14:40:53

©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：整体题目不难，但是比较麻烦。主要麻烦的点在于（1）正则表达式上，即如何接收这样tuple的数据；（2）算中位数

代码

```
#
import re
n = int(input())
dis = input()
price = list(input().split())
price = [int(_) for _ in price]
dis = re.findall(r'\((\d+),(\d+)\)', dis)
dis = [int(x)+int(y) for x, y in dis]

c1 = [dis[_]/int(price[_]) for _ in range(n)]
sortc1 = sorted(c1)
sortprice = sorted(price)

def mid(list):
    nlen = len(list)
    if nlen % 2 == 0:
        res = (list[nlen//2] + list[nlen//2-1])/2
    else:
        res = list[nlen//2]

    return res

midc1 = mid(sortc1)
midprice = mid(sortprice)

count = 0
for i in range(n):
    if c1[i] > midc1 and price[i] < midprice:
        count +=1

print(count)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: **Accepted**

源代码

```
import re
n = int(input())
dis = input()
price = list(input().split())
price = [int(_) for _ in price]
dis = re.findall(r'\((\d+),(\d+)\)', dis)
dis = [int(x)+int(y) for x, y in dis]

c1 = [dis[_]/int(price[_]) for _ in range(n)]
sortc1 = sorted(c1)
sortprice = sorted(price)

def mid(list):
    nlen = len(list)
    if nlen % 2 == 0:
        res = (list[nlen//2] + list[nlen//2-1])/2
    else:
        res = list[nlen//2]

    return res

midc1 = mid(sortc1)
midprice = mid(sortprice)

count = 0
for i in range(n):
    if c1[i] > midc1 and price[i] < midprice:
        count +=1

print(count)
```

基本信息

#: 44133415
题目: 19963
提交人: 李佳霖2000013713
内存: 4680kB
时间: 27ms
语言: Python3
提交时间: 2024-03-09 15:23:32

27300: 模型整理

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

思路：同样也是思路简单，但很麻烦的一道题。希望期末考试不会有这种题

代码

```
#
import re

n = int(input())
models = {}
pattern = r"[+]?\\d*\\.\\d+([eE][+]?\\d+)?"

for i in range(n):
```



```

model, paNum = input().split('-') # model and parameter numbers
# num_match = re.search(pattern, paNum)
num = paNum[:-1]
# num = num_match.group()
if paNum[-1] == 'B':
    num = float(num) * 1e9
elif paNum[-1] == 'M':
    num = float(num) * 1e6

if model not in models:
    models[model] = []
models[model].append([num, paNum])

# models = {key: sorted(values) for key, values in models.items()}
sorted_models = sorted(models.items())

for model, parameters in sorted_models:
    parameters = sorted(parameters, key = lambda x: x[0]) # 参数量从小到大排序
    # parameters_str = ', '.join([f'{int(param):,d}' for param in parameters]) # 将参数量格式
    # 化为带有千位分隔符的字符串
    parameters_str = ', '.join([f'{param[1]}' for param in parameters]) # 将参数量格式化为带有
    # 千位分隔符的字符串
    print(f'{model}: {parameters_str}')

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
import re

n = int(input())
models = {}
pattern = r"[-+]?d*\.\d+([eE][-+]?d+)?"

for i in range(n):
    model, paNum = input().split('-') # model and parameter numbers
    # num_match = re.search(pattern, paNum)
    num = paNum[: -1]
    # num = num_match.group()
    if paNum[-1] == 'B':
        num = float(num) * 1e9
    elif paNum[-1] == 'M':
        num = float(num) * 1e6

    if model not in models:
        models[model] = []
    models[model].append([num, paNum])

# models = {key: sorted(values) for key, values in models.items()}
sorted_models = sorted(models.items())

for model, parameters in sorted_models:
    parameters = sorted(parameters, key = lambda x: x[0]) # 参数量从小到大
    # parameters_str = ', '.join([f'{int(param):,d}' for param in parameters])
    parameters_str = ', '.join([f'{param[1]}' for param in parameters])
    print(f'{model}: {parameters_str}')
```

基本信息

#: 44135074

题目: 27300

提交人: 李佳霖2000013713

内存: 3856kB

时间: 25ms

语言: Python3

提交时间: 2024-03-09 16:09:03

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。

这次作业主要回忆了dp和递归的写法，最后两道题练习了正则表达式的写法。