Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Complied by 李鹏辉，元培学院

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

Windows 10 Home, PyCharm 2022.3.2 (Community Edition)

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：递归。25分钟。

代码

```
def intercept(upper_bound, heights):
    possible_heights = [height for height in heights if height <=
upper_bound]
```

```
    if len(possible_heights) == 1:
        return 1
    if len(possible_heights) == 0:
        return 0
    return max(1+intercept(possible_heights[0], possible_heights[1:]),
intercept(upper_bound, heights[1:]))


def q1():
    ignore = input()
    heights = list(map(int, input().split()))
    print(intercept(1000000000, heights))


q1()
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def intercept(upper_bound, heights):
    possible_heights = [height for height in heights if height <= upper_
    if len(possible_heights) == 1:
        return 1
    if len(possible_heights) == 0:
        return 0
    return max(1+intercept(possible_heights[0], possible_heights[1:]), 
```

基本信息

#: 44171764
题目: 02945
提交人: 2100017777_李鹏辉(2100017777)
内存: 3616kB
时间: 26ms
语言: Python3
提交时间: 2024-03-11 18:16:21

04147:汉诺塔问题(Tower of Hanoi)

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

思路：递归。12分钟。

代码

```
def hanoi(n, source, target, bridge):
    if n == 1:
        print(f"1:{source}->{target}")
        return
    else:
        hanoi(n-1, source, bridge, target)
        print(f"{n}:{source}->{target}")
        hanoi(n-1, bridge, target, source)
        return


def q2():
    raw_list = input().split()
```

```
    hanoi(int(raw_list[0]), raw_list[1], raw_list[3], raw_list[2])


q2()
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def hanoi(n, source, target, bridge):
    if n == 1:
        print(f"1:{source}->{target}")
        return
    else:
        hanoi(n-1, source, bridge, target)
        print(f"{n}:{source}->{target}")
        hanoi(n-1, bridge, target, source)
        return
```

基本信息

#: 44171942
题目: 04147
提交人: 2100017777_李鹏辉(2100017777)
内存: 3544kB
时间: 22ms
语言: Python3
提交时间: 2024-03-11 18:32:31

03253: 约瑟夫问题No.2

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

思路：20分钟。

代码

```
def joseph(n, p, m):
    players = list(range(1, n+1))
    index = p - 1
    result = ''
    while len(players) > 1:
        index += m - 1
        index %= len(players)
        deleted = players.pop(index)
        result += str(deleted) + ','
    result += str(players[0])
    print(result)


def q3():
    while True:
        raw_string = input()
        if raw_string == '0 0 0':
            break
        inputs = tuple(map(int, raw_string.split()))
        joseph(*inputs)


q3()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
def joseph(n, p, m):
    players = list(range(1, n+1))
    index = p - 1
    result = ''
    while len(players) > 1:
        index += m - 1
        index %= len(players)
```

基本信息

#: 44173020
题目: 03253
提交人: 2100017777_李鹏辉(2100017777)
内存: 3636kB
时间: 22ms
语言: Python3
提交时间: 2024-03-11 19:30:07

21554:排队做实验 (greedy)v0.2

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

思路：15分钟。

代码

```
def q4():
    n = int(input())
    times = list(map(int, input().split()))
    for i in range(1, n+1):
        times[i-1] = [times[i-1], i]
    times.sort(key=lambda x: x[0])
    order = []
    sum_result = 0
    for j in range(1, n+1):
        order.append(times[j-1][1])
        sum_result += (n - j) * times[j-1][0]
    print(' '.join(map(str, order)))
    print("{:.2f}".format(sum_result / n))


q4()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
def q4():
    n = int(input())
    times = list(map(int, input().split()))
    for i in range(1, n+1):
        times[i-1] = [times[i-1], i]
    times.sort(key=lambda x: x[0])
    order = []
    sum_result = 0
```

基本信息

#: 44173912
题目: 21554
提交人: 2100017777_李鹏辉(2100017777)
内存: 3632kB
时间: 22ms
语言: Python3
提交时间: 2024-03-11 20:08:57

19963:买学区房

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

思路：30分钟。

代码

```
def q5_body(qualities, prices):
    def find_median(nums):
        sorted_nums = sorted(nums)
        if len(nums) % 2 == 0:
            max_middle = len(nums) // 2
            return (sorted_nums[max_middle] + sorted_nums[max_middle - 1]) / 2
        else:
            return sorted_nums[len(nums) // 2]

    median_quality = find_median(qualities)
    median_price = find_median(prices)
    result = 0
    for _ in range(len(qualities)):
        if qualities[_] > median_quality and prices[_] < median_price:
            result += 1
    print(result)


def q5():
    n = int(input())
    qualities = []
    distances = input().strip().split()
    prices = list(map(int, input().split()))
    for i in range(n):
        distance = distances[i]
        x, y = tuple(eval(distance))
        qualities.append((x + y) / prices[i])
    q5_body(qualities, prices)


q5()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
def q5_body(qualities, prices):
    def find_median(nums):
        sorted_nums = sorted(nums)
        if len(nums) % 2 == 0:
            max_middle = len(nums) // 2
            return (sorted_nums[max_middle] + sorted_nums[max_middle - 1
        else:
            return sorted_nums[len(nums) // 2]
```

基本信息

#: 44181428
题目: 19963
提交人: 2100017777_李鹏辉(2100017777)
内存: 4128kB
时间: 39ms
语言: Python3
提交时间: 2024-03-12 14:04:07

27300: 模型整理

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

思路：30分钟。

代码

```
def q6():
    n = int(input())
    model_dict = {}
    ori_mb = "{'M':[], 'B':[]}"
    for _ in range(n):
        model, size = tuple(input().split('-'))
        if model not in model_dict:
            model_dict[model] = dict(eval(ori_mb))
        size_num_raw = size[:-1]
        if '.' in size_num_raw:
            size_num = float(size_num_raw)
        else:
            size_num = int(size_num_raw)
        size_uni = size[-1]
        model_dict[model][size_uni].append(size_num)
    sorted_dict = sorted(model_dict.items())

    def print_dict(name, sizes):
        result = f"{name}: "
        m_sizes = sorted(sizes['M'])
        if m_sizes:
            for size in m_sizes:
                result += f'{size}M, '
        b_sizes = sorted(sizes['B'])
        if b_sizes:
            for size in b_sizes:
                result += f'{size}B, '
        print(result[:-2])

    for pair in sorted_dict:
        print_dict(pair[0], pair[1])


q6()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
def q6():
    n = int(input())
    model_dict = {}
    ori_mb = "{'M':[], 'B':[]}"
    for _ in range(n):
        model, size = tuple(input().split('-'))
        if model not in model_dict:
            model_dict[model] = dict(eval(ori_mb))
```

基本信息

#：44181173
题目：27300
提交人：2100017777_李鹏辉(2100017777)
内存：3660kB
时间：22ms
语言：Python3
提交时间：2024-03-12 13:30:50

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。

第5题用map(sum(eval()))的语法可以少写不少行。第6题的参考答案写法很漂亮，将million换算成billion比大小，同时将名义值与实际值同时储存，达到比较实际值、输出名义值的效果，比按M或B分别储存未经换算的实际值要方便得多。