Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Complied by ==张宇帆 心院==

编程环境

Python编程环境：Spyder IDE 5.2.2

1. 题目

02945: 拦截导弹

思路：本来觉得很难，因为之前接触过类似的题目TLE了，所以不怎么敢写，考试也就这道没做出来（约瑟夫的除外！）。结果用递归就可以过了（之前接触的不可以），后来小小地改了一下用了dp，发现和群里大佬给的代码基本一样，非常开心的。

代码

```
#
n = int(input())
hig = list(map(int,input().split()))
dp = [1]*n
result = 0
for i in range(n-2,-1,-1):
    for j in range(i+1, n):
        if hig[i] >= hig[j]:
            dp[i] = max(dp[i], dp[j]+1)
    result = max(result, dp[i])
print(result)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n = int(input())
hig = list(map(int,input().split()))
dp = [1]*n
result = 0
for i in range(n-2,-1,-1):
    for j in range(i+1, n):
        if hig[i] >= hig[j]:
            dp[i] = max(dp[i], dp[j]+1)
    result = max(result, dp[i])
print(result)
```

基本信息

#: 44096244
题目: 02945
提交人: 2200013720
内存: 3584kB
时间: 26ms
语言: Python3
提交时间: 2024-03-06 20:12:19

04147:汉诺塔问题(Tower of Hanoi)

思路：简单的dp，不过一开始在减小问题规模那块卡住了，思索了十多分钟才想明白
用fr和to表示从fr到to处，然后每次需要经过：从fr把num-1个盘子挪到另外一个（也就是3-fr-to），然后把编号为n的盘子从fr挪到to，然后从3-fr-to挪num-1个盘子到to处即可

代码

```
#
num, A, B, C = input().split()
position = [A, B, C]
num = int(num)
def move(num, fr, to):
    if num == 1:
        print("%s:%s->%s"%(num, position[fr], position[to]))
    else:
        move(num-1, fr, 3-fr-to)
        print("%s:%s->%s"%(num, position[fr], position[to]))
        move(num-1, 3-fr-to, to)
move(num, 0, 2)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
num, A, B, C = input().split()
position = [A, B, C]
num = int(num)
def move(num, fr, to):
    if num == 1:
        print("%s:%s->%s"%(num, position[fr], position[to]))
    else:
        move(num-1, fr, 3-fr-to)
        print("%s:%s->%s"%(num, position[fr], position[to]))
        move(num-1, 3-fr-to, to)
move(num, 0, 2)
```

基本信息

#:	44090835
题目:	E04147
提交人:	2200013720
内存:	3536kB
时间:	21ms
语言:	Python3
提交时间:	2024-03-06 15:59:15

03253: 约瑟夫问题No.2

思路：用列表模拟队列即可，考试的时候下意识把result用了字符串，初始化为了”，导致当输入有两位数时WA了，这个错误在之前也有犯过，长记性了）

代码

```
#
n, start, gap = map(int,input().split())
while n != 0 and start != 0 and gap != 0:
    result = []
    queue = [i for i in range(1,n+1)]
    queue = queue[start-1:] + queue[:start-1]
    while len(queue) > 1:
```

```
        for m in range(gap):
            queue = queue[1:] + queue[:1]
        out = queue.pop()
        result.append(str(out))
    result.append(str(queue.pop()))
    print(','.join(result))
    n, start, gap = map(int,input().split())
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
n, start, gap = map(int,input().split())
while n != 0 and start != 0 and gap != 0:
    result = []
    queue = [i for i in range(1,n+1)]
    queue = queue[start-1:] + queue[:start-1]
    while len(queue) > 1:
        for m in range(gap):
            queue = queue[1:] + queue[:1]
        out = queue.pop()
        result.append(str(out))
    result.append(str(queue.pop()))
    print(','.join(result))
    n, start, gap = map(int,input().split())
```

基本信息

#:	44093688
题目:	03253
提交人:	2200013720
内存:	3636kB
时间:	20ms
语言:	Python3
提交时间:	2024-03-06 17:23:15

21554:排队做实验 (greedy)v0.2

思路：排序后让做实验时间少的同学先做即可

代码

```
#
n = int(input())
time = list(map(int,input().split()))
for i in range(n):
    time[i] = [time[i],i+1]
time.sort()
totime = 0
result = []
for i in range(n):
    if i == n-1:
        pass
    else:
        totime += time[i][0]*(n-i-1)
    result.append(str(time[i][1]))
print(' '.join(result))
print("%.2f"%(totime/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 = int(input())
time = list(map(int,input().split()))
for i in range(n):
    time[i] = [time[i],i+1]
time.sort()
totime = 0
result = []
for i in range(n):
    if i == n-1:
        pass
    else:
        totime += time[i][0]*(n-i-1)
    result.append(str(time[i][1]))
print(' '.join(result))
print("%.2f"%(totime/n))
```

基本信息

#: 44091714
题目: M21554
提交人: 2200013720
内存: 3616kB
时间: 21ms
语言: Python3
提交时间: 2024-03-06 16:27:59

19963:买学区房

思路：对性价比和价格排序然后取交集

代码

```
#
n = int(input())
pairs = [i[1:-1] for i in input().split()]
distances = [sum(map(int,i.split(','))) for i in pairs]
value = list(map(int,input().split()))
benefit = []
for i in range(n):
    benefit.append([distances[i]/value[i], i])
    value[i] = [value[i], i]
value.sort()
benefit.sort()
va = set()
be = set()
if n%2 == 0:
    midbe = (benefit[n//2-1][0]+benefit[n//2][0])/2
    midva = (value[n//2-1][0]+value[n//2][0])/2
else:
    midbe = benefit[n//2][0]
    midva = value[n//2][0]
for i in range(n//2):
    if benefit[n-i-1][0] > midbe:
        be.add(benefit[n-i-1][1])
    if value[i][0] < midva:
        va.add(value[i][1])
print(len(va.intersection(be)))
```

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())
pairs = [i[1:-1] for i in input().split()]
distances = [sum(map(int,i.split(','))) for i in pairs]
value = list(map(int,input().split()))
benefit = []
for i in range(n):
    benefit.append([distances[i]/value[i], i])
    value[i] = [value[i], i]
value.sort()
benefit.sort()
va = set()
be = set()
if n%2 == 0:
    midbe = (benefit[n//2-1][0]+benefit[n//2][0])/2
    midva = (value[n//2-1][0]+value[n//2][0])/2
else:
    midbe = benefit[n//2][0]
    midva = value[n//2][0]
for i in range(n//2):
    if benefit[n-i-1][0] > midbe:
        be.add(benefit[n-i-1][1])
    if value[i][0] < midva:
        va.add(value[i][1])
print(len(va.intersection(be)))
```

基本信息

#: 44092075
题目: T19963
提交人: 2200013720
内存: 5080kB
时间: 28ms
语言: Python3
提交时间: 2024-03-06 16:39:37

27300: 模型整理

思路：取字典然后排序即可，过程中将B和M转化为对应单位比然后计算出实际数值即可

代码

```
#
n = int(input())
model = []
para = {}
havevisited = []
for _ in range(n):
    name, size = map(str,input().split('-'))
    model.append(name)
    if size[-1] == 'B':
        realsize = float(size[:-1])*1000
    else:
        realsize = float(size[:-1])
    if name in para:
        para[name].append([realsize, size])
    else:
        para[name] = [[realsize, size]]
model.sort()
for m in model:
    if m not in havevisited:
        resize = sorted(para[m])
        result = [x[1] for x in resize]
        print("%s: %s"%(m, ', '.join(result)))
        havevisited.append(m)
```

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())
model = []
para = {}
havevisited = []
for _ in range(n):
    name, size = map(str,input().split('-'))
    model.append(name)
    if size[-1] == 'B':
        realsize = float(size[:-1])*1000
    else:
        realsize = float(size[:-1])
    if name in para:
        para[name].append([realsize, size])
    else:
        para[name] = [[realsize, size]]
model.sort()
for m in model:
    if m not in havevisited:
        resize = sorted(para[m])
        result = [x[1] for x in resize]
        print("%s: %s"%(m, ', '.join(result)))
        havevisited.append(m)
```

基本信息

#: 44092816
题目: T27300
提交人: 2200013720
内存: 3664kB
时间: 22ms
语言: Python3
提交时间: 2024-03-06 16:59:05

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。==

这次考试感觉状态不是很好，拦截导弹那道就发了一个小时呆，最后幡然醒悟连忙做后面的，以至于最后3分钟才交上最后一道。论实际难度其实确实不是很大，是考试时的状态不太多，有点担心到时候在机房考试了。

不管怎样，还是得好好练习，目前的计划是把dp再好好练练，然后开始回顾dfs和bfs，树的话也可以找一些基础题做做，总的就三方面，哪方面做累了就想想另一方面。

在这周课程中提到了八皇后和滑雪，我回去重新做了一遍，看了之前的代码感觉惨不忍睹，因为都不知道变量代表的是啥，于是重新写了段并详细标了下注释，希望能给第一次接触的同学们一点帮助）

八皇后

```
## 每次queen函数的执行结束都是为numlist添加元素，是对numlist的直接修改
def queen(startboard, numlist, row = 0):
    ## startboard储存各行皇后排列，未排列的位置放-1（即原题中的a），第i位储存第i行皇后的列数
    ## numlist储存最后排列
    ## 结束条件：当所有排列都完成后，所遍历的行必然为startboard的长度，此时直接储存即可
    if len(startboard) == row:
        ## 由于row和col从0开始，故最终的数据应当每位都加上1
        numlist.append(int(''.join(map(str,startboard))) + 11111111)
    else:
        ## 第row行皇后所在行的遍历（按列进行）
        for col in range(len(startboard)):
            ## 第row行皇后在第col列
            ## flag作为检测是否符合要求的指标
            startboard[row] = col
            flag = True
            ## 检查这个row行col列皇后是否符合要求（以flag为指标），prerow表示先前第prerow行的皇后
            ## or前语句：判断是否与之前有过的第cul行皇后同列；or后语句：判断是否在对角
```

```
            ## 若不符，则计算第row行皇后是否可以在下一列
            ## 若相符，则计算下一行皇后可以在哪一列
            for prerow in range(row):
                if col == startboard[prerow] or (abs(row - prerow) == abs(col -
startboard[prerow])):
                    flag = False
                    break
            if flag:
                queen(startboard, numlist, row + 1)
result = []
queen([-1]*8, result)
n = int(input())
for _ in range(n):
    t = int(input())
    print(result[t-1])
```

滑雪

```
order = [[-1,0], [1,0], [0,-1], [0,1]] ## 滑雪动作（方向）
dp = [] ## 储存路径地图
Map = [] ## 储存高度地图
## 每次ski函数的执行必然返回对应单元格的最长子路径
def ski(row, col, dp, order, Map):
    ## 遇到已经找到了最长路径的单元格则直接返回
    if dp[row][col] > 1:
        return dp[row][col]
    else:
        for ac in order:
            ## 判断会不会滑出地图
            if row+ac[0] < 0 or row+ac[0] >= len(Map) or col+ac[1] < 0 or
col+ac[1] >= len(Map[0]):
                pass
            ## 路径+1条件，即可不可以滑过去
            elif Map[row][col] > Map[row+ac[0]][col+ac[1]]:
                ## 每个动作遍历一遍，从上下左右找到最长路径然后+1
                dp[row][col] = max(dp[row][col], ski(row+ac[0], col+ac[1], dp,
order, Map)+1)
            ## 每个动作遍历后返回找到的最长路径，如果上述elif不成立，即它是四周最低的，那就返
回初始值1
        return dp[row][col]

R, C = map(int,input().split())
result = 0
for _ in range(R):
    Map.append(list(map(int,input().split())))
    dp.append([1]*C)
## 对每个单元格遍历一遍，找到最长路径即可
for row in range(R):
    for col in range(C):
        ## 对求得的每个单元格的最长路径的集合取最大值
        result = max(result, ski(row, col, dp, order, Map))
print(result)
```