

Assignment #B: Dec Mock Exam大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Compiled by 任宇桐 物理学院

说明：

- 1) 月考：AC1 (请改为同学的通过数)。考试题目都在“题库（包括计概、数算题目）”里面，按照数字题号能找到，可以重新提交。作业中提交自己最满意版本的代码和截图。
- 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。

1. 题目

E22548: 机智的股民老张

<http://cs101.openjudge.cn/practice/22548/>

思路：

虽然是E级别难度题目，但是还是做了很久...主要原因似乎是没有tag，所以一开始不知道从什么地方下手，但是后来还是想到扫描两遍，记录前一段、后一段的最大最小值，分别进行比较。

代码：

```

a = list(map(int, input().split()))
ref = []
minv = 10001
for i in range(len(a)):
    minv = min(minv, a[i])
    ref.append(minv)
maxv = 0
ans = 0
for j in range(len(a)-1, -1, -1):
    maxv = max(maxv, a[j])
    ans = max(ans, maxv-ref[j])

print(ans)

```

代码运行截图（至少包含有"Accepted"）

#47566693提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

a = list(map(int, input().split()))
ref = []
minv = 10001
for i in range(len(a)):
    minv = min(minv, a[i])
    ref.append(minv)
maxv = 0
ans = 0
for j in range(len(a)-1, -1, -1):
    maxv = max(maxv, a[j])
    ans = max(ans, maxv-ref[j])

print(ans)

```

基本信息

#: 47566693
 题目: E22548
 提交人: 24n2400011498
 内存: 9584kB
 时间: 62ms
 语言: Python3
 提交时间: 2024-12-05 15:22:23

M28701: 炸鸡排

greedy, <http://cs101.openjudge.cn/practice/28701/>

思路:

虽然是M级别的难度题目，但是个人感觉还是有点难，当然也可能是greedy学的实在不好... 试图总结一下思路，首先确定最优的情形，之所以炸鸡排会出现停止的情形是由于某鸡排所需时间长导致的，不能将其分配到其他炸鸡锅继续炸，也就是说，如果鸡排炸不完，那一定是由于锅是空的导致的，所以找到了问题的限制条件，出现了greedy的突破口。

代码:

```
n, k = map(int, (input().split()))
s = list(map(int, input().split()))
s.sort()
total = sum(s)
max_time = sum(s)/k
if s[-1]>max_time:
    for i in range(n-1, -1, -1):
        if s[i] <= max_time:
            break
        k -= 1
        total -= s[i]
        max_time = total/k
print('%.3f' % max_time)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n, k = map(int, (input().split()))
s = list(map(int, input().split()))
s.sort()
total = sum(s)
max_time = sum(s)/k
if s[-1]>max_time:
    for i in range(n-1, -1, -1):
        if s[i] <= max_time:
            break
        k -= 1
        total -= s[i]
        max_time = total/k
print('%.3f' % max_time)
```

基本信息

#: 47578468
题目: 28701
提交人: 24n2400011498
内存: 3624kB
时间: 21ms
语言: Python3
提交时间: 2024-12-05 21:52:30

M20744: 土豪购物

dp, <http://cs101.openjudge.cn/practice/20744/>

思路:

开始其实想到了dp, 但是一直考虑在dp的最后找到答案, 所以一直没能搞定...后来参考了群内大家的讨论才勉强搞定, 想起来原来可以考虑“以...结尾的序列”和双重dp。

代码:

```
s = list(map(int, input().split(',')))
n = len(s)
dp1 = [-float('inf')]*(n+1)
dp2 = [-float('inf')]*(n+1)
for i in range(1,n+1):
    dp1[i] = max(dp1[i-1]+s[i-1], s[i-1])
    dp2[i] = max(dp1[i-1], dp2[i-1]+s[i-1], dp1[i])
print(max(dp2))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
s = list(map(int, input().split(',')))
n = len(s)
dp1 = [-float('inf')]*(n+1)
dp2 = [-float('inf')]*(n+1)
for i in range(1,n+1):
    dp1[i] = max(dp1[i-1]+s[i-1], s[i-1])
    dp2[i] = max(dp1[i-1], dp2[i-1]+s[i-1], dp1[i])
print(max(dp2))
```

基本信息

#: 47621212
题目: 20744
提交人: 24n2400011498
内存: 9440kB
时间: 70ms
语言: Python3
提交时间: 2024-12-08 11:48:25

T25561: 2022决战双十一

brute force, dfs, <http://cs101.openjudge.cn/practice/25561/>

思路:

此题考试的时候花了很多时间处理,但是最红未能搞定, 后来检查考场上写出的代码,发现整体思路没有问题,错误仅仅出现在字典元素的使用。发现这种基础知识如果出错非常难以检查出来,一定要平时总结好记住。(虽然之前写在了cheatsheet上但是考试的时候根本想不起来是这个地方出错...)

代码:

```
n, m = map(int, input().split())
prices = [dict() for _ in range(n)]
for i in range(n):
```

```

s = list(input().split())
for j in range(len(s)):
    separate_idx = s[j].index(':')
    market = int(s[j][:separate_idx])-1
    price = int(s[j][separate_idx+1:])
    prices[i][market] = price

coupons = []
for k in range(m):
    t = input().split()
    temp = []
    for l in range(len(t)):
        separate_idx = t[l].index('-')
        q = int(t[l][:separate_idx])
        x = int(t[l][separate_idx+1:])
        temp.append((q, x))
    temp.sort()
    coupons.append(temp[:])

min_total_bill = float('inf')
shopping_list = [0 for _ in range(m)] #从m个店铺购买的商品总价
def dfs(kind):
    global min_total_bill
    if kind == n:
        total_bill = 0
        for i in range(m):
            money = shopping_list[i]
            real_cost = money
            for coupon in coupons[i]:
                if money >= coupon[0]:
                    real_cost = min(real_cost, money-coupon[1])
            else:
                break
        total_bill += real_cost
        times = sum(shopping_list)//300
        total_bill -= times*50
        min_total_bill = min(total_bill, min_total_bill)
    return

for market, price in prices[kind].items():
    shopping_list[market] += price
    dfs(kind+1)

```

```
shopping_list[market] -= price
dfs(0)
print(min_total_bill)
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
n, m = map(int, input().split())
prices = [dict() for _ in range(n)]
for i in range(n):
    s = list(input().split())
    for j in range(len(s)):
        separate_idx = s[j].index(':')
        market = int(s[j][:separate_idx])-1
        price = int(s[j][separate_idx+1:])
        prices[i][market] = price

#print(prices)

coupons = []
for k in range(m):
    t = input().split()
    temp = []
    for l in range(len(t)):
        separate_idx = t[l].index('-')
        q = int(t[l][:separate_idx])
        x = int(t[l][separate_idx+1:])
```

基本信息

#: 47574661
题目: 25561
提交人: 24n2400011498
内存: 3720kB
时间: 32ms
语言: Python3
提交时间: 2024-12-05 18:43:55

T20741: 两座孤岛最短距离

dfs, bfs, <http://cs101.openjudge.cn/practice/20741/>

思路:

这道题目其实基本思路很好想，但是想要优化到不超时还是需要下功夫的。首先，只需要将找到的第一个孤岛打上标记，第二个不用管。其次，可以直接在找到第一个孤岛之后直接开始bfs，不用再次遍历列表了。

代码:

```
from collections import deque
import sys
sys.setrecursionlimit(1<<30)

n = int(input())
maze = []
```

```

for _ in range(n):
    s = input()
    m = len(s)
    maze.append(s)

visited1 = [[False]*m for _ in range(n)]

def scope(x, y):
    return 0<=x<n and 0<=y<m

def dfs(x, y, visited):
    visited[x][y] = True ### 这个顺序可以保证初始点被直接打上标记
    for dx, dy in [(1, 0), (-1, 0), (0, -1), (0, 1)]:
        nx = x + dx
        ny = y + dy
        if scope(nx, ny) and not visited[nx][ny] and maze[nx][ny]
== '1':
            dfs(nx, ny, visited)

def bfs(x, y, step):
    q = deque([(x, y, step)])
    inqueue = {(x, y)}

    while q:
        x, y, step = q.popleft()
        for dx, dy in [(0, 1), (0, -1), (-1, 0), (1, 0)]:
            nx = x + dx
            ny = y + dy
            if scope(nx, ny) and maze[nx][ny] == '1' and not
visited1[nx][ny]:
                return step
            if scope(nx, ny) and (nx, ny) not in inqueue:
                if maze[nx][ny] == '1':
                    q.appendleft((nx, ny, step))
                elif maze[nx][ny] == '0':
                    q.append((nx, ny, step+1))
            inqueue.add((nx, ny))

def find_1():
    for i in range(n):
        for j in range(m):

```

```

        if maze[i][j] == '1':
            dfs(i, j, visited1)
        return bfs(i, j, 0)

print(find_1())

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

from collections import deque
import sys
sys.setrecursionlimit(1<<30)

n = int(input())
maze = []
for _ in range(n):
    s = input()
    m = len(s)
    maze.append(s)

visited1 = [[False]*m for _ in range(n)]

def scope(x, y):
    return 0<=x<n and 0<=y<m

def dfs(x, y, visited):
    visited[x][y] = True ### 这个顺序可以保证初始点被直接打上标记
    for dx, dy in [(1, 0), (-1, 0), (0, -1), (0, 1)]:
        nx = x + dx
        ny = y + dy
        if scope(nx, ny) and not visited[nx][ny] and maze[nx][ny] == '1':
            dfs(nx, ny, visited)

def bfs(x, y, step):

```

基本信息

#: 47578034
 题目: 20741
 提交人: 24n2400011498
 内存: 4968kB
 时间: 37ms
 语言: Python3
 提交时间: 2024-12-05 21:27:07

T28776: 国王游戏

greedy, <http://cs101.openjudge.cn/practice/28776>

思路:

第一次看到题目实在不知道是在干什么...看了题解, 尝试理解了一下, 发现似乎只有思维上的难度...尝试写一下思路, 自己梳理一下:

考虑一个已经排好最小值的队列 s , 这等价于交换两个大臣, 最大值不会减小。容易发现, 不包含最大值的交换都不会造成影响。不妨设含有最大值的这两个编号分别为 $i, i+1$:

$$money[i] = pre // b[i], money[i+1] = pre * a[i] // b[i+1]$$

交换以后有:

$$money[i+1] = pre // b[i+1], money[i] = pre * a[i+1] // b[i]$$

容易看出：

$$money[i+1] \geq money[i], money[i+1] \geq money[i]$$

即 $(i+1)$ 大臣拿到的变少， i 大臣拿到的变多。如果 i 大臣本来就拿的多，那自然满足条件，所以只需要考虑 i 大臣拿的少的情况。只需要要求交换后 i 大臣拿的比原来 $i+1$ 大臣拿的多即可保证 s 为最小排列：

$$money[i] \geq money[i+1]$$

化简，容易看出这等价于：

$$a[i] * b[i] \leq a[i+1] * b[i+1]$$

代码：

```
n = int(input())
a, b = map(int, input().split())
s = []
for _ in range(n):
    x, y = (map(int, input().split()))
    s.append((x*y, x, y))
s.sort()
pre = a
maxv = 0
for i in range(n):
    maxv = max(maxv, pre//s[i][2])
    pre *= s[i][1]
print(maxv)
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
n = int(input())
a, b = map(int, input().split())
s = []
for _ in range(n):
    x, y = map(int, input().split())
    s.append((x*y, x, y))
s.sort()
pre = a
maxv = 0
for i in range(n):
    maxv = max(maxv, pre//s[i][2])
    pre *= s[i][1]
print(maxv)
```

基本信息

#: 47575423
题目: 28776
提交人: 24n2400011498
内存: 3612kB
时间: 23ms
语言: Python3
提交时间: 2024-12-05 19:24:40

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“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

这次月考做的很不理想，一方面可能和考试状态有关，一开始没有预料到考试难度骤增，直接打乱了做题节奏，每个题目都想研究一下，但是几乎都没有成功解决，另一方面发现做题的思维还是过于死板，虽然每日选做也完成了不少，但是没有仔细思考代码的逻辑，所以考试的时候很难快速迁移到考题当中。下来以后尝试自己思考发现还是能多搞定一些题目的，希望接下来能回顾一下之前完成题目找找感觉。