

Assignment #B: Dec Mock Exam大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Compiled by 颜鼎堃 工学院

说明:...

- 1) 月考: AC2 (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:

- 没啥好说的, 只要找每一个当前最小值之后的最大值就好
- 可以用 `min()` 再优化一下, 不过无所谓了

代码:

```
1  a = list(map(int, input().split()))
2  max_prof = 0
3  min_price = 1e9
4  for i in a:
5      if i < min_price:
6          min_price = i
7      if i - min_price > max_prof:
8          max_prof = i - min_price
9  print(max_prof)
```

Python

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 24n2400011125 信箱 账号

CS101 / 20241205 cs101 Mock Exam大雪前一天 已经结束

题目 排名 状态 统计 提问

#47566475提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
a = list(map(int, input().split()))
max_prof = 0
min_price = 1e9
for i in a:
    if i < min_price:
        min_price = i
    if i - min_price > max_prof:
        max_prof = i - min_price
print(max_prof)
```

基本信息

#: 47566475
题目: E22548
提交人: 颜鼎盛(24n2400011125)
内存: 9560kB
时间: 33ms
语言: Python3
提交时间: 2024-12-05 15:15:30

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

M28701: 炸鸡排

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

思路:

- 第一个要点是给出以下断言: 所有鸡排能够全部煎熟, 一点都不浪费的条件是

$$\frac{1}{k} \sum_{i=1}^n t_i \geq \max\{t_i\}$$

- 易知, 如果不满足上式, 至少所需时间最长的鸡排一定煎不熟, 于是我们有以下第二个断言: 假如不满足第一个条件, 那么不断去除时间最长的鸡排, 直至满足

$$\frac{1}{k-r} \sum_{i=1}^{n-r} t_i \geq \max\{t_i\}$$

为止, 此时最长时间即为

$$\frac{1}{k-r} \sum_{i=1}^{n-r} t_i$$

其中 r 为去除掉的鸡排数量, $r < k$

显然总是可以达到这一状态, 因为当 $r = k - 1$ 条件必然成立

代码:

```
1 n, k = map(int, input().split())
2 chicken = sorted(map(int, input().split()))
3 for i in range(k):
```

Python

```

4     if sum(chicken) / (k - i) < chicken[-1]:
5         chicken.pop()
6     else:
7         print(f"{sum(chicken) / (k - i):.3f}")
8         break

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#47637388提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

n, k = map(int, input().split())
chicken = sorted(map(int, input().split()))
for i in range(k):
    if sum(chicken) / (k - i) < chicken[-1]:
        chicken.pop()
    else:
        print(f"{sum(chicken) / (k - i):.3f}")
        break

```

基本信息

#:

47637388

题目:

28701

提交人:

颜鼎堃(24n2400011125)

内存:

3612kB

时间:

21ms

语言:

Python3

提交时间:

2024-12-09 10:50:49

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M20744: 土豪购物

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

思路:

- 考场上没想到用动规，考场上的想法是对于数列中每一个负数，将其改为0，然后求前缀和，再化归为第一题，这样的算法复杂度为 $O(n^2)$ ，判我超时

代码:

Python

```

1  val = list(map(int, input().split(",")))
2  n = len(val)
3  dp = [[0 for i in range(n + 1)] for j in range(2)]
4  dp[0][1] = val[0]
5  dp[0][2] = max(val[0] + val[1], val[1])
6  dp[1][2] = val[1]
7  for i in range(2, 1 + n):
8      dp[0][i] = max(dp[0][i - 1] + val[i - 1], val[i - 1])
9      dp[1][i] = max(dp[1][i - 1] + val[i - 1], dp[0][i - 2] + val[i - 1])
10 print(max(map(max, dp)))
11

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#47620239提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
val = list(map(int, input().split(",")
n = len(val)
dp = [[0 for i in range(n + 1)] for j in range(2)]
dp[0][1] = val[0]
dp[0][2] = max(val[0] + val[1], val[1])
dp[1][2] = val[1]
for i in range(2, 1 + n):
    dp[0][i] = max(dp[0][i - 1] + val[i - 1], val[i - 1])
    dp[1][i] = max(dp[1][i - 1] + val[i - 1], dp[0][i - 2] + val[i - 1])
print(max(map(max, dp)))
```

基本信息

#:

47620239

题目:

20744

提交人:

颜鼎堃(24n2400011125)

内存:

9756kB

时间:

77ms

语言:

Python3

提交时间:

2024-12-08 10:52:40

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

T25561: 2022决战双十一

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

思路:

- 遍历一遍,但是这题遍历都不好遍历
- 准确来说不是深搜,只是个递归罢了,但确实挺麻烦

代码:

1

def dfs(num_goods, pref, n):

2

if num_goods == n:

3

global min_price

4

tot_tag = 0

5

tot_off = 0

6

for i in range(m):

7

max_off = 0

8

tagged = sum(pref[i])

9

for j in coupon[i]:

10

if tagged >= j:

11

max_off = max(max_off, coupon[i][j])

12

tot_tag += tagged

13

tot_off += max_off

14

tot_off += tot_tag // 300 * 50

15

min_price = min(min_price, tot_tag - tot_off)

16

else:

17

for i in goods[num_goods]:

18

pref[i - 1].append(goods[num_goods][i])

Python

```

19         dfs(num_goods + 1, pref, n)
20         pref[i - 1].pop()
21
22
23 n, m = map(int, input().split())
24 goods = [input().split() for i in range(n)]
25 coupon = [input().split() for i in range(m)]
26 for i in range(n):
27     for j in range(len(goods[i])):
28         goods[i][j] = map(int, goods[i][j].split(":"))
29     goods[i] = dict(goods[i])
30 for i in range(m):
31     for j in range(len(coupon[i])):
32         coupon[i][j] = map(int, coupon[i][j].split("-"))
33     coupon[i] = dict(coupon[i])
34
35 min_price = 1e9
36 dfs(0, [[] for i in range(m)], n)
37 print(min_price)
38

```

代码运行截图（至少包含有"Accepted"）

#47649523提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def dfs(num_goods, pref, n):
    if num_goods == n:
        global min_price
        tot_tag = 0
        tot_off = 0
        for i in range(m):
            max_off = 0
            tagged = sum(pref[i])
            for j in coupon[i]:
                if tagged >= j:
                    max_off = max(max_off, coupon[i][j])
            tot_tag += tagged
            tot_off += max_off
        tot_off += tot_tag // 300 * 50
        min_price = min(min_price, tot_tag - tot_off)
    else:
        for i in goods[num_goods]:
            pref[i - 1].append(goods[num_goods][i])
            dfs(num_goods + 1, pref, n)
            pref[i - 1].pop()

n, m = map(int, input().split())
goods = [input().split() for i in range(n)]
coupon = [input().split() for i in range(m)]
for i in range(n):
    for j in range(len(goods[i])):
        goods[i][j] = map(int, goods[i][j].split(":"))
    goods[i] = dict(goods[i])
for i in range(m):
    for j in range(len(coupon[i])):
        coupon[i][j] = map(int, coupon[i][j].split("-"))
    coupon[i] = dict(coupon[i])

min_price = 1e9
dfs(0, [[] for i in range(m)], n)
print(min_price)

```

基本信息

#: 47649523

题目: 25561

提交人: 颜鼎堃(24n2400011125)

内存: 3700KB

时间: 34ms

语言: Python3

提交时间: 2024-12-09 20:33:14

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

T20741: 两座孤岛最短距离

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

思路:

- 学语法千日用语法一时，`yield` 表达式真好使
- 把两个岛的边界找出来，存起来，找曼哈顿距离最小值，可过

- 可以转成集合避免重复计算耽误时间

代码：

```
1 DIRECTIONS = ((0, 1), (0, -1), (1, 0), (-1, 0))
2 def dfs(x, y):
3     stack = [(x, y)]
4     while stack:
5         x1, y1 = stack.pop()
6         isl[x1][y1] = "0"
7         for dx, dy in DIRECTIONS:
8             nx, ny = x1 + dx, y1 + dy
9             if isl[nx][ny] == "1":
10                 stack.append((nx, ny))
11             else:
12                 yield (x1, y1)
13
14
15 n = int(input())
16 isborder = []
17 cnt = 0
18 isl = ["0" for i in range(n+2)] + [["0"] + list(input()) + ["0"] for i in
19 range(n)] + ["0" for i in range(n+2)]
20 for i in range(1, 1 + n):
21     while "1" in isl[i]:
22         cnt += 1
23         isborder.append(set(dfs(i, isl[i].index("1"))))
24         if cnt == 2:
25             break
26     if cnt == 2:
27         break
28 print(min([abs(a - c) + abs(b - d) - 1 for a, b in isborder[0] for c, d in
29 isborder[1]]))
```

Python

代码运行截图 (至少包含有"Accepted")



#47568904提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

DIRECTIONS = ((0, 1), (0, -1), (1, 0), (-1, 0))
def dfs(x, y):
    stack = [(x, y)]
    while stack:
        x1, y1 = stack.pop()
        isl[x1][y1] = "0"
        for dx, dy in DIRECTIONS:
            nx, ny = x1 + dx, y1 + dy
            if isl[nx][ny] == "1":
                stack.append((nx, ny))
            else:
                yield (x1, y1)

n = int(input())
isborder = []
cnt = 0
isl = [["0" for i in range(n+2)]] + [["0"] + list(input()) + ["0"] for i in range(1, 1 + n)]
for i in range(1, 1 + n):
    while "1" in isl[i]:
        cnt += 1
        isborder.append(set(dfs(i, isl[i].index("1"))))
        if cnt == 2:
            break
    if cnt == 2:
        break
print(min([abs(a - c) + abs(b - d) - 1 for a, b in isborder[0] for c, d

```

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

基本信息

#: 47568904
 题目: T20741
 提交人: 颜鼎堃(24n2400011125)
 内存: 11692kB
 时间: 223ms
 语言: Python3
 提交时间: 2024-12-05 16:28:35

T28776: 国王游戏

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

思路:

- 最开始看错题了，以为是一直乘到当前大臣的左手再除以当前大臣的右手，但是因祸得福，这种情况比题目中情况还简单些。先考虑这种情况，以两个大臣左右手分别写上 $(a, b), (c, d)$ 为例，注意到

$$\frac{a}{b} \leq \frac{ac}{b}, \frac{c}{d} \leq \frac{ac}{d}$$

故只要按照右手上数字大小排序，就可以得到含有最大值的最小值的序列了，求出这个序列的最大值即可

- 对于题目中的情况，还是相同的道理，但是由于最后每一个排列的最后一个数分子不相同，故分子分母同时乘上当前大臣的左手上的数，得

$$\frac{1}{l_n r_n} \prod_{i=1}^n$$

故按照 $l_i r_i$ 排序即可

代码:

```

1  n = int(input())
2  a, b = map(int, input().split())
3  hands = sorted([list(map(int, input().split())) for i in range(n)], key=lambda
    t: t[1] * t[0])
4  min_max_awa = 0
5  for i in range(1, n):
6      a *= hands[i - 1][0]
```

Python

```
7     min_max_awa = max(min_max_awa, a // hands[i][1])
8     print(min_max_awa)
```

代码运行截图（至少包含有"Accepted"）



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。

考完考试的那一天彻底破防了，但是也尽量靠自己完成了所有的作业题目

如果作业题目简单 ← 每次我看到这几个字我都很无语

等下，怎么我刚写完作业新作业就发了