Assignment #5: 20251009 cs101 Mock Exam寒露第二天

Updated 1651 GMT+8 Oct 9, 2025

2025 fall, Complied by 郭旭杰、化学与分子工程学院

> **说明：**
>
> 1. **解题与记录：**
>
> 对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. 提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E29895: 分解因数

implementation, http://cs101.openjudge.cn/practice/29895/

耗时：10min

思路：较简单，10min速通。

需要注意的一点是不能直接找n最大的真因数，否则会TLE，而是用n整除其除了1以外最小的因数以节省时间。

代码

```
# Python3
n = int(input())
i = 2
while n % i != 0:
    i += 1
print(n // i)
```

代码运行截图 （至少包含有"Accepted"）

E29940: 机器猫斗恶龙

greedy, http://cs101.openjudge.cn/practice/29940/

耗时：15min

思路：较简单，15min速通。

简单的贪心算法，就是假设机器猫的血量可以为负值或0，一开始的血量为0，然后将其血量的变化“平移”，使其最小值为1，输出“平移”的距离即可。

代码

```
n = int(input())
js = list(map(int,input().split()))
sgm = 0
s = []
for j in js:
    sgm += j
    s.append(sgm)
m = min(s)
if m < 0:
    res = 1-m
else:
    res = 1
print(res)
```

代码运行截图 （至少包含有"Accepted"）

29940:机器猫斗恶龙

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

描述

机器猫出门斗恶龙了！他需要通过 n 个关卡。

每个关卡要么是与怪物战斗，扣除一定的血量；要么是营地，给机器猫增加一定的血量。

在旅途中，机器猫**任意时刻**的血量不能低于或等于 0。问机器猫至少需要多少的初始血量，才能完成任务。

血量为正整数。

输入

第一行，一个正整数 n，表示关卡数量。

第二行，共 n 个整数 a_i，表示每个关卡。

- 若 a_i>0，则表示这个关卡是营地，增加 a_i 的血量
- 若 a_i<0，则表示这个关卡是战斗，机器猫血量代价为 a_i

所有数据满足 n <= 100000, 1 <= |a_i| <= 1000。

输出

仅一行，一个正整数，表示机器猫需要的初始血量。

样例输入

```
sample1 input:
3
-100 -200 -300

sample1 output:
601
```

全局题号 **29940**

添加于 **2025-10-09**

提交次数 **111**

尝试人数 **73**

通过人数 **73**

Other language verions

English

你的提交记录

#	结果	时间
1	Accepted	2025-10-09

M29917: 牛顿迭代法

implementation, http://cs101.openjudge.cn/practice/29917/

耗时：前前后后加起来一个多小时。

思路：这题不难，照着题目意思写代码就可以了。但是我肯定不可能在考试中将其做对。

只因我当时对不定行输入一无所知。一直要么WA要么RE。

不定行输入：

while True:

try :

代码

except:

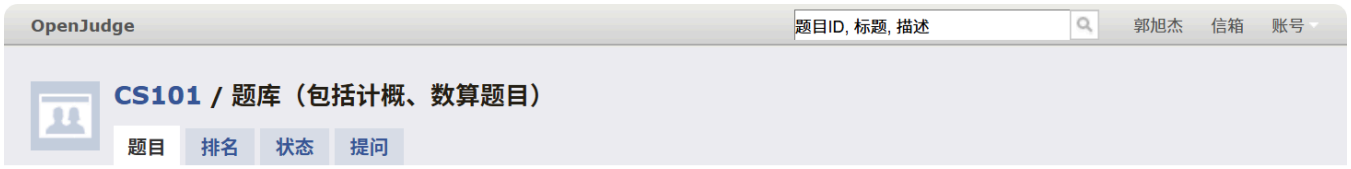
break

代码

```
while True:
    try:
        t = float(input())
        x = 1
        y = 1 - t
        k = 0
        while abs(y/(2*x)) > 1e-6:
            x -= y / (2*x)
            y = x**2 - t
            k += 1
```

```
        x -= y / (2*x)
        k += 1
        print(k,"%.2f" % x)
    except:
        break
```

代码运行截图 （至少包含有"Accepted"）



29917:牛顿迭代法

查看 | 提交 | 统计 | 提问

总时间限制: 1000ms　内存限制: 65536kB

全局题号 **29917**
添加于 **2025-10-09**
提交次数 **362**
尝试人数 **112**
通过人数 **104**

你的提交记录

#	结果	时间
1	Accepted	2025-10-10

描述

用牛顿法求一个数的平方根。

要求:
初始取 1 为近似根，因此第一次迭代是过曲线上横坐标为 1 的点作切线。
迭代终止条件为，上一次迭代求出的近似根，与这一次迭代求出的更精确的近似根的差，小于等于 1E-6。
输出迭代终止时的迭代次数。

迭代公式:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

输入

多行，每行可能是一个整数或一个小数。

输出

对输入的每行，输出一行，为对应的迭代次数和近似根，用空格分隔。

迭代次数为一个整数。
根为一个浮点数，保留小数点后两位。

样例输入

```
12
25
144
```

样例输出

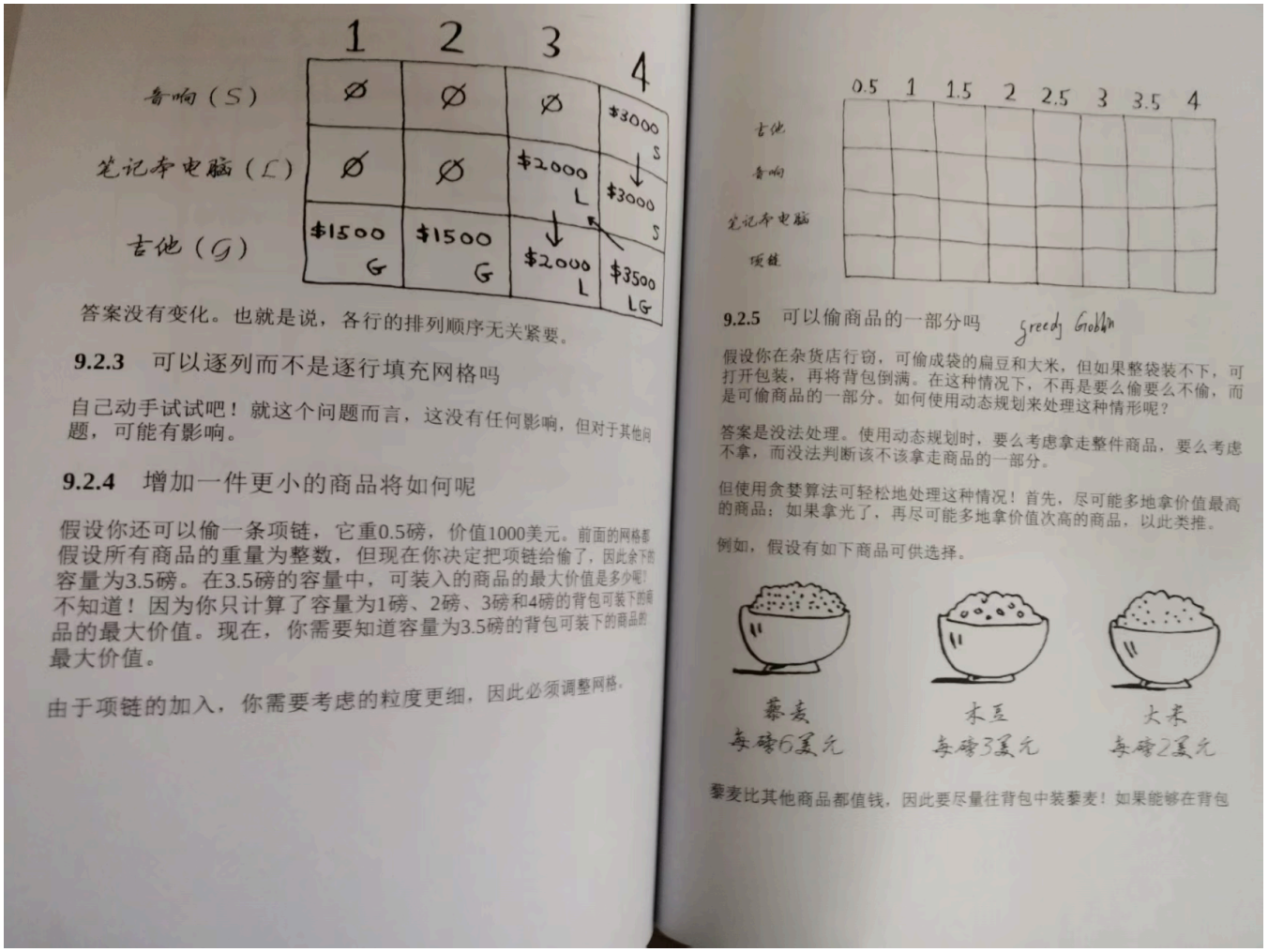
M29949: 贪婪的哥布林

greedy, http://cs101.openjudge.cn/practice/29949/

耗时：看书学会思路，不断优化终于成功。

思路：需要对二维数组有较灵活的运用能力，有一定难度。考试时候对二维数组理解不深，总是想用Hash Table，就没能在考试期间完成。

这题是我在看完《算法图解》之后完成的。



答案没有变化。也就是说，各行的排列顺序无关紧要。

9.2.3 可以逐列而不是逐行填充网格吗

自己动手试试吧！就这个问题而言，这没有任何影响，但对于其他问题，可能有影响。

9.2.4 增加一件更小的商品将如何呢

假设你还可以偷一条项链，它重0.5磅，价值1000美元。前面的网格都假设所有商品的重量为整数，但现在你决定把项链给偷了，因此余下的容量为3.5磅。在3.5磅的容量中，可装入的商品的最大价值是多少呢？不知道！因为你只计算了容量为1磅、2磅、3磅和4磅的背包可装下的商品的最大价值。现在，你需要知道容量为3.5磅的背包可装下的商品的最大价值。

由于项链的加入，你需要考虑的粒度更细，因此必须调整网格。

9.2.5 可以偷商品的一部分吗

假设你在杂货店行窃，可偷成袋的扁豆和大米，但如果整袋装不下，可打开包装，再将背包倒满。在这种情况下，不再是要么偷要么不偷，而是可偷商品的一部分。如何使用动态规划来处理这种情形呢？

答案是没法处理。使用动态规划时，要么考虑拿走整件商品，要么考虑不拿，而没法判断该不该拿走商品的一部分。

但使用贪婪算法可轻松地处理这种情况！首先，尽可能多地拿价值最高的商品；如果拿光了，再尽可能多地拿价值次高的商品，以此类推。

例如，假设有如下商品可供选择。

藜麦 每磅6美元　　扁豆 每磅3美元　　大米 每磅2美元

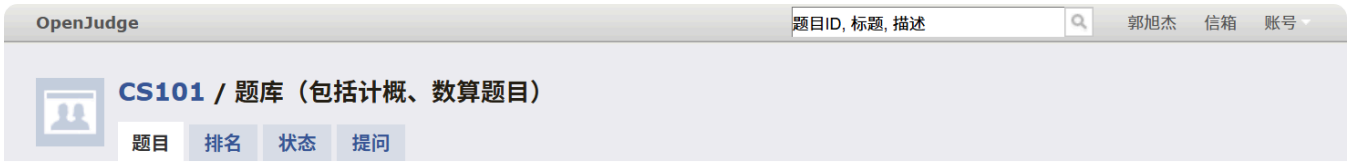
藜麦比其他商品都值钱，因此要尽量往背包中装藜麦！如果能够在背包

《算法图解》详细讲述了用动态规划处理“小偷问题”，其中略提及的用贪婪算法处理“偷商品的一部分”和这题几乎一模一样。就是按照矿石的价格排序，依次拿走，直至矿石被拿完或背包装满。

代码

```
#
n,m = map(int, input().split())
js = []
sgm = 0
earn = 0
for _ in range(n):
    v,w = map(int, input().split())
    js.append((v / w,v,w))
js.sort(reverse=True)
for j in js:
    if sgm + j[2] <= m:
        sgm += j[2]
        earn += j[1]
    else:
        earn += j[0] * (m - sgm)
        sgm = m
        break
print("%.2f" % earn)
```

代码运行截图 （至少包含有"Accepted"）



29949:贪婪的哥布林

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

全局题号 **29949**

添加于 **2025-10-09**

提交次数 **175**

尝试人数 **88**

通过人数 **85**

你的提交记录

#	结果	时间
3	Accepted	2025-10-12
2	Runtime Error	2025-10-11
1	Runtime Error	2025-10-11

描述

艾泽拉斯的一只地精（Goblin，哥布林）意外地发现了一个废弃的矿洞，里面有数堆不同的矿石（例如铋矿、镔爪矿、亚基矿等）。每一堆矿石都有其总价值和总重量。

哥布林的背包最多只能装下总重量为 W 的物品。幸运的是，这些矿石都是粉末状的，他可以从任何一堆矿石中取出任意一部分（例如取走一堆铋矿的 1/3）。

作为一名贪婪的哥布林，他希望背包里装的矿石总价值尽可能高。请你帮他计算出这个最大总价值。

输入

第一行包含两个整数 N 和 M (1 <= N <= 100, 1 <= M <= 10000)，分别代表矿石的堆数和背包的载重上限。
接下来 N 行，每行包含两个整数 v 和 w (1 <= v, w <= 1000)，分别代表这堆矿石的总价值和总重量。

输出

输出一个浮点数，代表能装入背包的最大总价值。结果保留两位小数。

样例输入

```
3 50
60 10
100 20
120 30
```

样例输出

```
240.00
```

提示

M29918: 求亲和数

implementation, http://cs101.openjudge.cn/practice/29918/

耗时：陆续做了一个下午和一个晚上

思路：这题很难了，考试时我想要先用较慢的代码输出所有结果，代码后台运算的时候我先做其它题目，然后对这一题暴力讨论。

结果我的代码复杂度太高，n取最大值的时候Pycharm楞是后台运算了一个小时也只输出两组结果，只能暂且放弃。

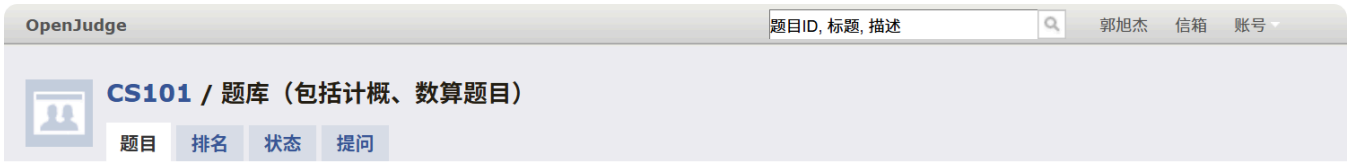
后来在讨论中我看到了其他同学的算法，这种算法有大量诸如将n的讨论缩减到讨论到n**0.5+1等优化，大大缩减了运行时间，提高了效率。

代码

```
#我的O(n**3)的代码，当n=100000时，一小时只能输出两组亲和数。
def sumf(x):
    sumf_r = 0
    for _ in range(1, x):
        if x % _ == 0:
            sumf_r += _
    return sumf_r

n = int(input())
for i in range(1, n+1):
    for j in range(i+1, n+1):
        if sumf(i) == j and sumf(j) == i:
```

```
            print(i,j)
            continue

#============================================================
#同学提供的一种快得多的算法O(nlogn)，虽然还是会TLE。
t = int(input())
n = [0]
for i in range(2,100001):
    d = 0
    for j in range(1,int(i**0.5)+1):
        if i % j == 0:
            if j ** 2 == i or j == 1:
                d += j
            else:
                d += j + i // j
    n.append(d)
for i in range(2,100001):
    if n[i-1] > 99999:
        continue
    elif n[n[i-1]-1]== i:
        if i < n[i-1] <= t:
            print(i,n[i-1])

#============================================================
#最终AC使用的方法：先用较快算法算出结果，然后就像下面这么做。这种算法O(1),实打实的brute force。
n = int(input())
if n >= 284:
    print(220,284)
if n >= 1210:
    print(1184,1210)
if n >= 2924:
    print(2620,2924)
if n >= 5564:
    print(5020,5564)
if n >= 6368:
    print(6232,6368)
if n >= 10856:
    print(10744,10856)
if n >= 14595:
    print(12285,14595)
if n >= 18416:
    print(17296,18416)
if n >= 76084:
    print(63020,76084)
if n >= 66992:
    print(66928,66992)
if n >= 71145:
    print(67095,71145)
if n >= 87633:
    print(69615,87633)
if n >= 88730:
    print(79750,88730)
```

代码运行截图 （至少包含有"Accepted"）



CS101 / 题库（包括计概、数算题目）

题目 排名 状态 提问

29918:求亲和数

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

全局题号 29918
添加于 2025-10-09
提交次数 411
尝试人数 102
通过人数 99

你的提交记录

#	结果	时间
3	Accepted	2025-10-09
2	Time Limit Exceeded	2025-10-09
1	Wrong Answer	2025-10-09

描述

遥远的古代，人们发现某些自然数之间有特殊的关系：设有两个数a和b，若a的所有除本身以外的因数之和等于b，b的所有除本身以外的因数之和等于a，则称a、b是一对亲和数。

例如220和284：
220的真因数包括：1,2,4,5,10,11,20,22,44,55,110.
1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284
284的真因数包括：1,2,4,71,142.
1 + 2 + 4 + 71 + 142 = 220
所以220和284是一对亲和数。

输入

一个正整数n，1<=n<=100000。

输出

所有亲和数对"a b"，满足a和b均小于等于n。
每个亲和数对占一行，两个数之间用一个空格隔开，较小数在前，较大数在后。
对于多个亲和数对，以较小数递增的顺序输出它们。

样例输入

```
1500
```

样例输出

```
220 284
1184 1210
```

提示

```python
def sumf(x):
    sumf_r = 0
    for _ in range(1, x):
        if x % _ == 0:
            sumf_r += _
    return sumf_r

n = int(input())
for i in range(1, n+1):
    for j in range(i+1, n+1):
        if sumf(i) == j and sumf(j) == i:
            print(i,j)
            continue
```

#50282560提交状态

查看 提交 统计 提问

状态: Time Limit Exceeded

源代码

```
t = int(input())
n = [0]
for i in range(2,100001):
    d = 0
    for j in range(1,int(i**0.5)+1):
        if i % j == 0:
            if j ** 2 == i or j == 1:
                d += j
            else:
                d += j + i // j
    n.append(d)
for i in range(2,100001):
    if n[i-1] > 99999:
        continue
    elif n[n[i-1]-1]== i:
        if i < n[i-1] <= t:
            print(i,n[i-1])
```

基本信息

#:	50282560
题目:	29918
提交人:	2500011906
内存:	7280kB
时间:	2358ms
语言:	Python3
提交时间:	2025-10-09 20:56:06

©2002-2022 POJ 京ICP备20010980号-1

T29947:校门外的树又来了（选做）

http://cs101.openjudge.cn/practice/29947/

耗时：从10月9日到10月12日，陆续做了4天，不断修改终于成功。

思路：

代码

```
n,line = list(map(int, input().split()))
tree = n + 1
cut = []
for _ in range(line):
    a,b = list(map(int, input().split()))
    cut.append((a,b))
cut.sort(key=lambda x:x[0])

init = 0
kil = 0
```

```
while init <= line-1:
    at = cut[init][0]
    bt = cut[init][1]
    while init < line-1 and cut[init+1][0] <= bt:
        init += 1
        bt = max(cut[init][1], cut[init-1][1],bt)
    init += 1
    kil += bt - at + 1

print(tree - kil)
```

代码运行截图 （至少包含有"Accepted"）

OpenJudge 题目ID, 标题, 描述 郭旭杰 信箱 账号

CS101 / 题库（包括计概、数算题目）

题目 排名 状态 提问

#50325390提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n,line = list(map(int, input().split()))
tree = n + 1
cut = []
for _ in range(line):
    a,b = list(map(int, input().split()))
    cut.append((a,b))
cut.sort(key=lambda x:x[0])

init = 0
kil = 0
while init <= line-1:
    at = cut[init][0]
    bt = cut[init][1]
    while init < line-1 and cut[init+1][0] <= bt:
        init += 1
        bt = max(cut[init][1], cut[init-1][1],bt)
    init += 1
    kil += bt - at + 1

print(tree - kil)
```

基本信息

#: 50325390
题目: 29947
提交人: 郭旭杰
内存: 3636kB
时间: 19ms
语言: Python3
提交时间: 2025-10-12 16:07:44

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

2. 学习总结和收获

通过这次作业，我学会了基本的贪心算法（E29895: 分解因数；E29940: 机器猫斗恶龙，M29949: 贪婪的哥布林；T28701炸鸡排）重点掌握了超时的优化方法（E29895: 分解因数；M29918: 求亲和数；T29947:校门外的树又来了）学会了不定行输入（M29917: 牛顿迭代法）对二维数组有了更深刻的认识（M29949: 贪婪的哥布林；T29947:校门外的树又来了）

这次考试让我看到了我的真实水平，还需要加强练习。

如果作业题目简单，有否额外练习题目，比如：OJ“计概2025fall每日选做”、CF、LeetCode、洛谷等网站题目。

周五夜间参加CodeForces比赛做出2题，九千多名，仍然是新手(Newbie)。

9839 LittleBeetroot 1133 488 00:06 645 00:35 -5

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

Problems

#	Name		
A	Circle of Apple Trees	standard input/output 1 s, 256 MB	x20318
B	Bitwise Reversion	standard input/output 1.5 s, 256 MB	x17938
C	Symmetrical Polygons	standard input/output 2 s, 256 MB	x8306
D	Not Alone	standard input/output 2 s, 256 MB	x3882
E	Zero Trailing Factorial	standard input/output 3 s, 512 MB	x550
F	Odd Queries on Odd Array	standard input/output 10 s, 1024 MB	x102

Complete problemset

Codeforces Round 1057 (Div. 2)

Finished

Practice

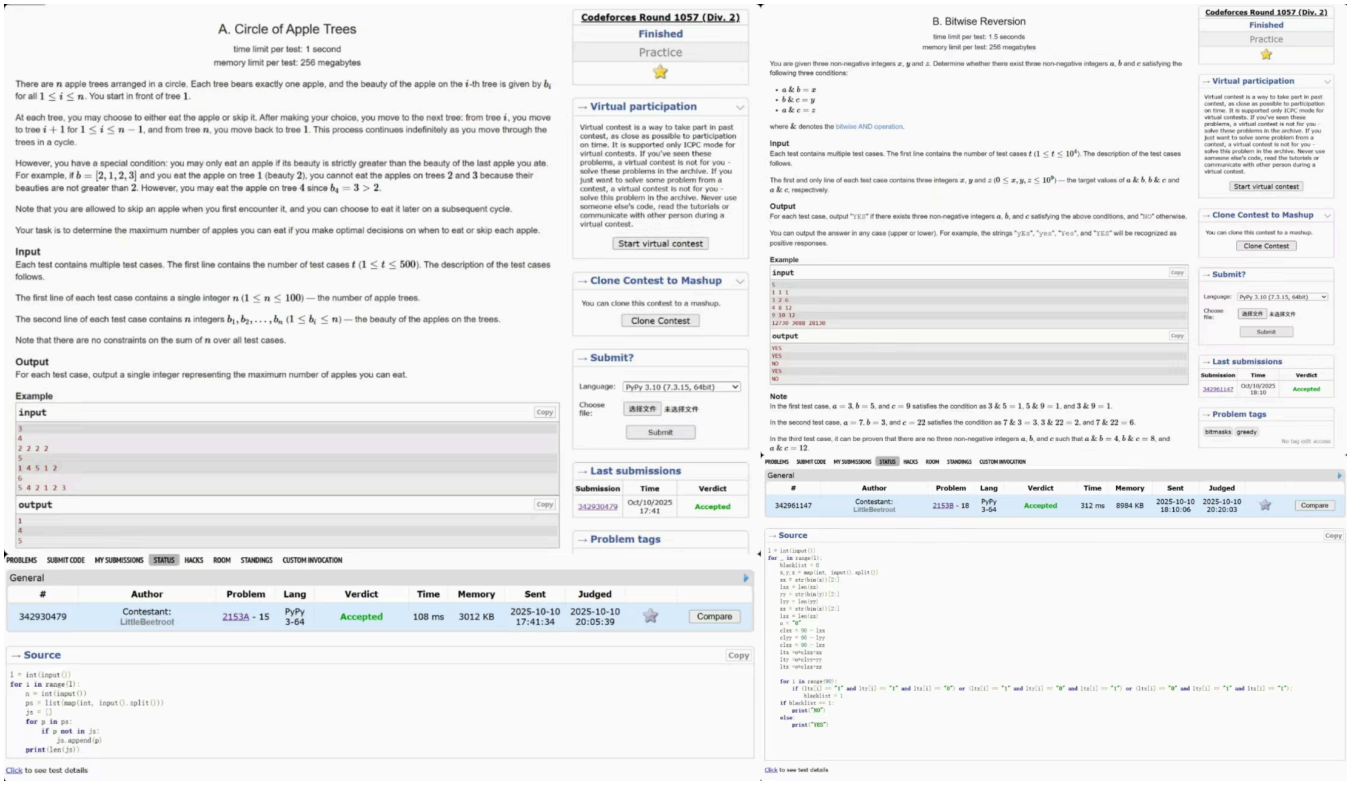
→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use

Codeforces Hot News!

Wow! Coder LittleBeetroot competed in Codeforces Round 1057 (Div. 2) and gained +219 rating points taking place 9311

Share it!



用在《算法图解》上学习的递归函数完成了“炸鸡排”这一道OJ上较难的题目。

惊讶地发现：用"%.3f" % result就会报错，用f"{result:.3f}"就能AC。


OpenJudge
题目ID, 标题, 描述
郭旭杰 信箱 账号
CS101 / 题库（包括计概、数算题目）
题目 排名 状态 提问
28701:炸鸡排
查看 提交 统计 提问
总时间限制: 1000ms 内存限制: 65536kB
描述
小P买了n块鸡排，想将它们做成美味的炸鸡排，其中第i块鸡排需要t[i]秒炸熟。小P只有一个炸锅，炸锅内可以放置k(k≤n)块鸡排。小P是个完美主义者，他要求任意时刻炸锅内必须恰有k块鸡排。他可以在任意时刻改变锅内正在炸的鸡排，只需保证已经熟了的鸡排不能继续留在锅中。小P希望知道炸鸡排最多可以持续多少时间。
例如，小P的三块鸡排需要分别需要1,1,1的时间炸熟，炸锅内需要放置2块鸡排，那么他决定在第一个0.5秒炸1和2两块鸡排，第二个0.5秒炸2和3两块鸡排，第三个0.5秒炸1和3两块鸡排，共持续了1.5秒。
输入
第一行输入两个正整数n和k，k≤n
第二行输入n个正整数，代表n块鸡排分别需要炸熟的时间t[1],t[2]...t[n]
输入数据保证，n≤1000，0<t[i]≤1000000
输出
输出一个双精度浮点数，代表炸鸡排最多可以持续的时间，结果保留三位小数。
样例输入
4 2
5 1 1 2
样例输出
4.000
第1秒，放置1和2
第2秒，放置1和3
第3、4秒，放置1和4
至此，2，3，4已经全部熟了，无法继续进行
全局题号 28701
添加于 2024-12-05
提交次数 470
尝试人数 191
通过人数 184
Other language verions
English
你的提交记录
结果 时间
3 Accepted 2025-10-12
2 Wrong Answer 2025-10-12
1 Wrong Answer 2025-10-12


OpenJudge
题目ID, 标题, 描述
郭旭杰 信箱 账号
CS101 / 题库（包括计概、数算题目）
题目 排名 状态 提问
#50326518提交状态
查看 提交 统计 提问
状态: Wrong Answer
源代码
def zhajipai(js,k):
js.sort(reverse=True)
res = sum(js) / k
if js[0] <= res:
return res
else:
new_js = js[1:]
nk = k-1
return zhajipai(new_js,nk)
n,k0 = map(int,input().split())
js0 = list(map(int,input().split()))
print(zhajipai(js0,k0))
基本信息
#: 50326518
题目: 28701
提交人: 郭旭杰
内存: 3500kB
时间: 22ms
语言: Python3
提交时间: 2025-10-12 16:55:01
©2002-2022 POJ 京ICP备20010980号-1
English 帮助 关于


OpenJudge
题目ID, 标题, 描述
郭旭杰 信箱 账号
CS101 / 题库（包括计概、数算题目）
题目 排名 状态 提问
#50326686提交状态
查看 提交 统计 提问
状态: Accepted
源代码
def zhajipai(js,k):
js.sort(reverse=True)
res = sum(js) / k
if js[0] <= res:
return res
else:
new_js = js[1:]
nk = k-1
return zhajipai(new_js,nk)
n,k0 = map(int,input().split())
js0 = list(map(int,input().split()))
print(f"{zhajipai(js0,k0):.3f}")
基本信息
#: 50326686
题目: 28701
提交人: 郭旭杰
内存: 6104kB
时间: 27ms
语言: Python3
提交时间: 2025-10-12 17:01:04
©2002-2022 POJ 京ICP备20010980号-1
English 帮助 关于