

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

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

28701:炸鸡排

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只有一个炸鸡锅，炸鸡锅内可以放置k(k≤n)块鸡排。小P是个完美主义者，他要求任意时刻炸鸡锅内必须有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，炸鸡锅内需要放置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 $t[1], t[2], \dots, t[n]$
输入数据保证， $n \leq 1000, 0 < t[i] \leq 1000000$

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
```

OpenJudge

题目ID, 标题, 描述

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 shaizipai(ja,k):
    ja.sort(reverse=True)
    res = sum(ja) / k
    if ja[0] <= res:
        return res
    else:
        new_ja = ja[1:]
        nk = k-1
        return shaizipai(new_ja,nk)

n,k = map(int,input().split())
ja = list(map(int,input().split()))
print(shaizipai(ja,k))
```

2000-2022 P01 01CPB20010900号-1 English 帮助 关于

OpenJudge

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 shaizipai(ja,k):
    ja.sort(reverse=True)
    res = sum(ja) / k
    if ja[0] <= res:
        return res
    else:
        new_ja = ja[1:]
        nk = k-1
        return shaizipai(new_ja,nk)

n,k = map(int,input().split())
ja = list(map(int,input().split()))
print(f'{shaizipai(ja,k):.3f}')
```

2000-2022 P01 01CPB20010900号-1 English 帮助 关于

基本信息

#: 50326518
题目: 28701
提交人: 陈旭杰
内存: 3500kB
时间: 22ms
语言: Python3
提交时间: 2025-10-12 16:55:01