

Assignment #9: Mock Exam立冬前一天

Updated 1658 GMT+8 Nov 6, 2025

2025 fall, Complied by 郭旭杰、化学与分子工程学院

说明:

1. Nov月考: AC4 (AC4) 。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. 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。
3. 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。
4. 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。

考试情况速览:



CS101 / 20251106 cs101 Mock Exam 立冬前一天 已经结束

题目 排名 状态 统计 提问

比赛已经结束

2025-11-06 15:08:00

开始时间

2025-11-06 17:00:00

结束时间

2025年11月计概 (2025fall-cs101: Algo DS-11班) 课程模拟考试。

请独立完成, 不能通讯, 如: 不能使用微信、邮件、QQ等工具。

考试期间, 请同学只访问OJ, 不能访问其他网站, 不要查看OJ考试之前自己提交的代码。

考试过程中允许可以带10张A4纸大小的cheat sheet, 以及空白草稿纸。

题目编号前面的大写字母, 相应表明是 Easy/Medium/Tough 级别。

登录别人的账号即视为违纪甚至作弊。把自己的账号密码告诉别人, 被别人登录, 也视为违纪甚至作弊。如果考前别人用过你的账号, 请立即修改密码。

请把你的昵称改为 25nxxxxx, 后面部分是学号。 <http://cs101.openjudge.cn/mine>
有同学昵称24n, 23n, ..., 19n开始也是可以的, 学号别错, 才能找到你的成绩。

题目ID	标题	通过率	通过人数	尝试人数
✓ E29982	一种等价类划分问题	81%	93	115
✓ E30086	dance	94%	111	118
✓ M25570	洋葱	91%	104	114
✓ M28906	数的划分	70%	58	83
— M29896	购物	48%	29	61
— T25353	排队	29%	7	24

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

52		25n2500011906(Little(25n2500011906)	4	05:35:22	00:15:36 (-2)	00:18:58	00:42:36 (-8)	00:58:12	(-5)	(-4)
----	--	-------------------------------------	---	----------	------------------	----------	------------------	----------	------	------

名次	名字	通过数	罚时	E29982	E30086	M25570	M28906	M29896	T25353
51	Ronny	4	05:09:57	00:30:00 (-5)	00:34:51	00:44:46 (-1)	01:20:20		
52	25n2500011906(Little(25n2500011906)	4	05:35:22	00:15:36 (-2)	00:18:58	00:42:36 (-8)	00:58:12	(-5)	(-4)

1. 题目

E29982:一种等价类划分问题

hashing, <http://cs101.openjudge.cn/practice/29982>

耗时: 15min, 3次尝试

思路: 去逗号加逗号要熟练, 基础的二维数组运用, 二维数组(mkf)分行分划预留足9999防止RE。

代码

```

def kcj(listy):
    return ','.join(listy)

m, n, k = map(int, input().split(','))
mkf = [[] for _ in range(9999)]
for i in range(m + 1, n):
    ts = i // 1000 + i % 1000 // 100 + i % 100 // 10 + i % 10
    if ts % k == 0:
        mkf[ts // k].append(str(i))

for row in mkf:
    if len(row) > 0:
        print(kcj(row))

```

代码运行截图 (至少包含有"Accepted")

E29982:一种等价类划分问题

总时间限制: 1000ms 内存限制: 65536kB

描述

在两个正整数m和n给定的整数范围内 (m 小于 n, 且不包括m和n) 取出各位数字之和均为k的倍数的所有数(k为正整数), 然后将这些数划分成若干个子集合, 每个子集合中的元素满足其各位数字之和相等, 请输出各个子集合, 其中 n 不大于10000。

每个集合元素按从小到大输出, 逗号间隔, 如果有多个集合, 则输出多行; 各位数字之和较小的集合在前面行输出。

例如, m=11, n=35, k=3

则,

12,21,30 这三个数的每位数字之和均为3, 且为3的倍数

15,24,33 这三个数的每位数字之和为6, 且为3的倍数

18,27 这两个数的每位数字之和为9, 也为3的倍数

由于三组数的最小数分别是12,15,18, 于是, 输出结果应为:

12,21,30

15,24,33

18,27

输入

一行, 三个值: m,n,k, 以逗号间隔

输出

输出各位数之和为k的倍数的若干行, 每一行中, 其元素的各位数字和相等, 且前面行元素的各位数字之和小于后面行元素的各位数字之和, 每行的元素按增序排列, 以逗号间隔。

样例输入

11,35,3



#50725912提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def kcj(listy):
    return ','.join(listy)

m, n, k = map(int, input().split(','))
mkf = [[] for _ in range(9999)]
for i in range(m + 1, n):
    ts = i // 1000 + i % 1000 // 100 + i % 100 // 10 + i % 10
    if ts % k == 0:
        mkf[ts // k].append(str(i))

for row in mkf:
    if len(row) > 0:
        print(kcj(row))
```

基本信息

#: 50725912
题目: E29982
提交人: 25n2500011906(Little(25n2500011906))
内存: 4388kB
时间: 24ms
语言: Python3
提交时间: 2025-11-06 15:23:36

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

E30086:dance

greedy, <http://cs101.openjudge.cn/practice/30086>

耗时: 3min, 1次通过

思路: 简单的送分题。

代码

```
n, d = list(map(int, input().split()))
js = list(map(int, input().split()))
js.sort()
blacklist = 0
for i in range(n):
    if js[2 * i + 1] - js[2 * i] > d:
        blacklist += 1
if blacklist > 0:
    print('No')
else:
    print('Yes')
```

代码运行截图 (至少包含有"Accepted")



CS101 / 20251106 cs101 Mock Exam 立冬前一天 已经结束

题目 排名 状态 统计 提问

E30086:dance

查看

提交

统计

提问

总时间限制: 1000ms 内存限制: 65536kB

描述

一所中学的一个班级有 $2N$ 名学生，第 $i (1 \leq i \leq 2N)$ 名学生的身高是 A_i 。

在即将开始的体育课上，学生们将两两配对，形成 N 组进行舞蹈。

要跳出优美的舞蹈，每一对舞伴之间的身高差必须小于或等于 D 。

给定每个学生的身高，确定能否跳出优美的舞蹈。

全局题号 30086

提交次数 188

尝试人数 118

通过人数 111

你的提交记录

#	结果	时间
1	Accepted	2025-11-06

输入

第一行输入两个整数 N, D 。

第二行输入 $2N$ 个整数 A_1, A_2, \dots, A_{2N} 。

输出

输出一行一个字符串，如果可以跳出优美的舞蹈输出 Yes，否则输出 No。

样例输入

```
6 4
22 15 32 36 16 30 42 30 39 23 17 18
```

样例输出

```
Yes
```

提示

考虑如下配对方案：



#50725990提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n, d = list(map(int, input().split()))
js = list(map(int, input().split()))
js.sort()
blacklist = 0
for i in range(n):
    if js[2 * i + 1] - js[2 * i] > d:
        blacklist += 1
if blacklist > 0:
    print('No')
else:
    print('Yes')
```

基本信息

#: 50725990
 题目: E30086
 提交人:
 25n2500011906(Little(25n2500011906))
 内存: 3608kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-11-06 15:26:58

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M25570: 洋葱

matrices, <http://cs101.openjudge.cn/practice/25570>

耗时: 24min,9次尝试

思路: 就是一层一层地剥洋葱直到剥完。

可以使用迭代或者递归, 这里由于n已经给出, 使用迭代 (for循环) 更方便易行。但是无论是那种方法都应该将最后两层单独讨论。

注意参数的取值等细节, 时间大多耗在这些细节上。

代码

```
def sgm_skin(amatrix):
    if len(amatrix) == 1:
        return amatrix[0][0]
    elif len(amatrix) == 2:
        return amatrix[0][0] + amatrix[0][1] + amatrix[1][0] + amatrix[1][1]
    else:
        k0 = 0
        k0 += sum(amatrix[0])
        k0 += sum(amatrix[-1])
        for _ in range(1, len(amatrix) - 1):
            k0 += amatrix[_][0]
            k0 += amatrix[_][-1]
        return k0

def peel(amatrix):
    if len(amatrix) <= 2:
```

```
        return []
    elif len(amatrix) == 3:
        return [[amatrix[1][1]]]
    else:
        new_amatrix = []
        for _ in range(1, len(amatrix) - 1):
            new_amatrix.append(amatrix[_][1:len(amatrix[_]) - 1])
        return new_amatrix

n = int(input())
js = []
matrix = []
for i in range(n):
    row = list(map(int, input().split()))
    matrix.append(row)
js.append(sgm_skin(matrix))
for j in range((n - 1)//2):
    matrix = peel(matrix)
    js.append(sgm_skin(matrix))

print(max(js))
```

代码运行截图 (至少包含有"Accepted")



M25570:洋葱

查看

提交

统计

提问

总时间限制: 1000ms 内存限制: 65536kB

描述

“如果你愿意一层一层一层地剥开我的心，你会发现，你会讶异，你是我最压抑最深处的秘密”

我们将洋葱抽象为一个 $n \times n$ 的矩阵，绕着矩阵最外侧环绕一圈，即得到一个“层”，然后将这个“层”中所有元素去掉，得到一个子矩阵，重复上述操作，即可得到若干个“层”。显然，若 n 为奇数，则位于其中心的那个元素也构成一个“层”。

现在给小明一个 $n \times n$ 的矩阵，小明想找到这个矩阵的“最大层”。“最大层”即为该矩阵每个“层”中数字之和最大的那个。

输入

第一行一个正整数 n ($n \leq 100$)

接下来 n 行，每行 n 个整数，代表矩阵中各个位置的数，均为不大于10000的非负整数

输出

一个整数，代表矩阵的“最大层”中的各数之和

样例输入

```
5
1 0 1 0 1
0 1 1 1 0
0 1 7 1 0
0 1 1 1 0
1 0 1 0 1
```

全局题号 **25570**提交次数 **226**尝试人数 **114**通过人数 **104**

Other language verions

[English](#)

你的提交记录

#	结果	时间
9	Accepted	2025-11-06
8	Runtime Error	2025-11-06
7	Wrong Answer	2025-11-06
6	Runtime Error	2025-11-06
5	Runtime Error	2025-11-06
4	Runtime Error	2025-11-06
3	Runtime Error	2025-11-06
2	Wrong Answer	2025-11-06
1	Wrong Answer	2025-11-06



#50726525提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def sgm_skin(amatrix):
    if len(amatrix) == 1:
        return amatrix[0][0]
    elif len(amatrix) == 2:
        return amatrix[0][0] + amatrix[0][1] + amatrix[1][0] + amatrix[1][1]
    else:
        k0 = 0
        k0 += sum(amatrix[0])
        k0 += sum(amatrix[-1])
        for _ in range(1, len(amatrix) - 1):
            k0 += amatrix[_][0]
            k0 += amatrix[_][-1]
        return k0

def peel(amatrix):
    if len(amatrix) <= 2:
        return []
    elif len(amatrix) == 3:
        return [[amatrix[1][1]]]
    else:
        new_amatrix = []
        for _ in range(1, len(amatrix) - 1):
            new_amatrix.append(amatrix[_][1:len(amatrix[_]) - 1])
        return new_amatrix

n = int(input())
js = []
matrix = []
for i in range(n):
    row = list(map(int, input().split()))
    matrix.append(row)
js.append(sgm_skin(matrix))
for j in range((n - 1)//2):
    matrix = peel(matrix)
    js.append(sgm_skin(matrix))

```

基本信息

#: 50726525
 题目: M25570
 提交人:
 25n2500011906(Little(25n2500011906))
 内存: 3972kB
 时间: 22ms
 语言: Python3

提交时间: 2025-11-06 15:50:36

M28906:数的划分

dfs, dp, <http://cs101.openjudge.cn/practice/28906>

耗时: 16min, 1次通过

思路: 典型的递归问题, 有一定难度, 要先在草稿纸上理清数学问题的思路再落实到代码上。

代码

```

def div(n0, k0):
    if k0 == 2:
        return n0 // 2
    else:
        res = 0
        i = 0
        while n0 - i * k0 - 1 >= 2:
            res += div(n0 - i * k0 - 1, k0 - 1)
            i += 1
    return res

n, k = map(int, input().split())
print(div(n, k))

```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 25n250011906 信箱 账号

CS101 / 20251106 cs101 Mock Exam立冬前一天 已经结束

题目 等级 状态 统计 提问

M28906:数的划分

查看 提交 统计 提问

全局题号 **28906**
提交次数 **166**
尝试人数 **83**
通过人数 **58**

总时间限制: 1000ms 内存限制: 65536kB

描述

将整数 n 分成 k 份，且每份不能为空，任意两个方案不相同（不考虑顺序）。

例如： $n=7, k=3$ ，下面三种分法被认为是相同的。

1,1,5;

1,5,1;

5,1,1.

问有多少种不同的分法。

Other language verions

English

你的提交记录

#	结果	时间
1	Accepted	2025-11-06

输入

一行两个整数 n, k ($6 < n \leq 200, 2 \leq k \leq 6$)，用空格隔开。

输出

1 个整数，即不同的分法。

样例输入

```

sample1 input:
7 3

sample1 output:
4

```

样例输出

```

sample2 input:
7 2

```



CS101 / 20251106 cs101 Mock Exam立冬前一天 已经结束

题目 排名 状态 统计 提问

#50726862提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def div(n0, k0):
    if k0 == 2:
        return n0 // 2
    else:
        res = 0
        i = 0
        while n0 - i * k0 - 1 >= 2:
            res += div(n0 - i * k0 - 1, k0 - 1)
            i += 1
        return res

n, k = map(int, input().split())
print(div(n, k))
```

基本信息

#: 50726862
 题目: M28906
 提交人:
 25n2500011906(Little(25n2500011906))
 内存: 3608kB
 时间: 28ms
 语言: Python3
 提交时间: 2025-11-06 16:06:12

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M29896:购物

greedy, <http://cs101.openjudge.cn/practice/29896>

耗时: 写了半个下午和一整个晚上, 最后求助AI并在群里讨论, 最终写出了自己的题解。

思路: 硬币面额从小到大凑, 但是较大面额的硬币面额尽量大, 这样才能使硬币总数最少, 这体现了贪心算法思想。

代码

```
x, n = list(map(int, input().split()))
js = list(map(int, input().split()))
# 输入数据

if 1 not in js:
    print('-1')
# 只有一种情况, 即没有一元硬币时, 会由于凑不出一元面值而不可能实现

else:
    js.sort(reverse=True)      # 倒序方便接下来找最大值(贪心)
    now_max = 0                # 现在持有的硬币最多可以全部表示从1到now_max的所有数
    cnt = 0                    # 需要的硬币总数为cnt

    while now_max < x:
        for j in js:
            if j <= now_max + 1:
                now_max += j      # 贪心: 找不大于now_max + 1的最大面额硬币,
# 这样能确保不存在不能被表示的数字。
                cnt += 1
```

```
break      # 确保每次都对所有硬币面额从大到小梳理。  
print(cnt)
```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 25n2500011906 信箱 账号 ▾

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

29896:购物

总时间限制: 1000ms 内存限制: 65536kB

描述

你就要去购物了, 现在你手上有 N 种不同面值的硬币, 每种硬币有无限多个。为了方便购物, 你希望带尽量少的硬币, 但要能组合出 1 到 X 之间的任意值。

输入

第一行两个数 X, N, 以下 N 个数, 表示每种硬币的面值。

输出

最少需要携带的硬币个数, 如果无解输出-1。

样例输入

```
20 4  
1 2 5 10
```

样例输出

```
5
```

提示

$N \leq 10, X \leq 10000$ (tag:greedy)

来源

[https://www.luogu.com.cn/problem/P1658\(TA-hhy\)](https://www.luogu.com.cn/problem/P1658(TA-hhy))

查看 提交 统计 提问

全局题号 **29896**
添加于 **2025-11-06**
提交次数 **261**
尝试人数 **92**
通过人数 **81**

你的提交记录

#	结果	时间
13	Accepted	2025-11-10
12	Accepted	2025-11-10
11	Wrong Answer	2025-11-10
10	Accepted	2025-11-10
9	Wrong Answer	2025-11-10
8	Accepted	2025-11-10
7	Accepted	2025-11-10
6	Wrong Answer	2025-11-10
5	Wrong Answer	2025-11-10
4	Wrong Answer	2025-11-10
3	Wrong Answer	2025-11-10
2	Accepted	2025-11-10
1	Wrong Answer	2025-11-10



CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#50784540提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
x, n = list(map(int, input().split()))
js = list(map(int, input().split()))
if 1 not in js:
    print('-1')
else:
    js.sort(reverse=True)
    now_max = 0
    cnt = 0
    while now_max < x:
        for j in js:
            if j <= now_max + 1:
                now_max += j
                cnt += 1
                break
    print(cnt)
```

基本信息

#: 50784540
 题目: 29896
 提交人:
 25n2500011906(Little(25n2500011906))
 内存: 3612kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-11-10 20:27:41

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于



CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#50785038提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
x, n = list(map(int, input().split()))
js = list(map(int, input().split()))
# 输入数据

if 1 not in js:
    print('-1')
# 只有一种情况, 即没有一元硬币时, 会由于凑不出一元面值而不可能实现

else:
    js.sort(reverse=True)      # 倒序方便接下来找最大值 (贪心)
    now_max = 0                # 现在持有的硬币最多可以全部表示从1到now_max的所有数
    cnt = 0                    # 需要的硬币总数为cnt

    while now_max < x:
        for j in js:
            if j <= now_max + 1:
                now_max += j      # 贪心: 找不大于now_max + 1的最大面额硬币,
    # 这样能确保不存在不能被表示的数字。
                cnt += 1
                break      # 确保每次都对所有硬币面额从大到小梳理。
    print(cnt)
```

基本信息

#: 50785038
 题目: 29896
 提交人:
 25n2500011906(Little(25n2500011906))
 内存: 3608kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-11-10 20:52:41

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

T25353:排队

greedy, <http://cs101.openjudge.cn/practice/25353>

耗时：写了大半个下午，最后不得不阅读题解。

思路：早就听说这一题排队特别难，这次终于可以会会它了，没想到我做不出来（TLE），求助AI结果现在几乎所有的AI(腾讯云宝，DeepSeek，豆包，夸克)也做不出来（WA）。最后还是看了老师提供的题解，有向无环图比较令人费解。

代码

自己的 (TLE)

```
# 我的代码（自己写的注释），应该正确，但是会严重超时
n, d = list(map(int, input().split()))
js = []
for i in range(n):
    j = int(input())
    js.append([j, i])
# 输入队列中所有人的身高和初始状态的序号
for i in range(n):
    pas = 0
    # pas变量表示这位同学最多还能往前排的位数
    while i - 1 - pas >= 0 and abs(js[i - 1 - pas][0] - js[i][0]) <= d:
        pas += 1
    # 求出pas的值
    if pas >= 1:
        for t in range(i - pas, i):
            if js[t][0] > js[i][0]:
                js[i][1] = js[t][1]
                # 如果后面同学的身高较低就尽可能往前面站
                for g in range(t, i):
                    js[g][1] += 1
                # 队列先不动，只是改变序号
                break
    js.sort(key=lambda x: x[1])
    # 按照序号重新排队，此时就能得到身高的字典序最小的队列
for j in js:
    print(j[0])
# 从前往后依次报出每一位同学的身高
```

题解提供 (AC)

```
# 25353: 排队
# 题意：每次只能交换相邻且身高差不超过 D 的人，求字典序最小的最终排列
# 思路：将允许交换关系看作有向无环图（DAG），每次取“入度为 0”的人中身高最小者输出。
# 实现：多轮扫描——每轮找出当前所有入度为 0 的节点集合 S,
# 对 S 内按身高升序输出并从序列中删除，重复直到序列为空。
# 注意：最坏情况下复杂度为 O(N^2)。
```

```

import sys
input = sys.stdin.readline

def solve():
    line = input().split()
    if not line:
        return
    n, D = map(int, line)
    arr = [int(input()) for _ in range(n)]

    result = []
    cur = arr[:] # 当前剩余队列

    while cur:
        m = len(cur)
        s_idx = [] # 当前入度为 0 的位置下标
        left_min = None
        left_max = None

        for i in range(m):
            h = cur[i]
            if i == 0:
                # 第一个人必然没有左侧约束, 入度为 0
                s_idx.append(i)
                left_min = h
                left_max = h
                continue

            # 判断是否满足: 与左侧所有人身高差均 ≤ D
            # 即 h 必须位于 [left_max - D, left_min + D] 区间内
            if left_max - D <= h <= left_min + D:
                s_idx.append(i)

            # 更新左侧区间最小/最大值
            if h < left_min:
                left_min = h
            if h > left_max:
                left_max = h

        # 收集并按身高升序排序
        s = [cur[i] for i in s_idx]
        s.sort()

        # 输出这些人
        result.extend(s)

        # 删除已输出的元素 (保持原相对顺序)
        to_remove = set(s_idx)
        cur = [cur[i] for i in range(m) if i not in to_remove]

    # 输出结果
    print('\n'.join(map(str, result)))

```

```
if __name__ == "__main__":
    solve()
```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 25n2500011906 信箱 账号

CS101 / 计算思维算法实践

题目 排名 状态 提问

T25353:排队

查看 提交 统计 提问

全局题号 **25353**
添加于 **2025-03-22**
提交次数 **229**
尝试人数 **45**
通过人数 **26**

描述

有 N 名同学从左到右排成一排, 第 i 名同学的身高为 h_i 。现在张老师想改变排队的顺序, 他能进行任意多次(包括0次)如下操作:

- 如果两名同学相邻, 并且他们的身高之差不超过 D, 那么老师就能交换他俩的顺序。

请你帮张老师算一算, 通过以上操作, 字典序最小的所有同学(从左到右)身高序列是什么?

输入

第一行包含两个正整数 N, D ($1 \leq N \leq 10^5$, $1 \leq D \leq 10^9$)。
接下去 N 行, 每行一个正整数 h_i ($1 \leq h_i \leq 10^9$) 表示从左到右每名同学的身高。

输出

输出 N 行, 第 i 行表示答案中第 i 名同学的身高。

样例输入

```
5 3
7
7
3
6
2
```

样例输出

```
6
7
7
2
```

你的提交记录

#	结果	时间
24	Accepted	2025-11-10
23	Accepted	2025-11-10
22	Runtime Error	2025-11-10
21	Runtime Error	2025-11-10
20	Runtime Error	2025-11-10
19	Wrong Answer	2025-11-10
18	Wrong Answer	2025-11-10
17	Wrong Answer	2025-11-10
16	Wrong Answer	2025-11-10
15	Wrong Answer	2025-11-10
14	Wrong Answer	2025-11-10
13	Wrong Answer	2025-11-10
12	Wrong Answer	2025-11-10
11	Wrong Answer	2025-11-10
10	Wrong Answer	2025-11-10
9	Wrong Answer	2025-11-10
8	Wrong Answer	2025-11-10
7	Time Limit Exceeded	2025-11-10
6	Time Limit Exceeded	2025-11-10
5	Wrong Answer	2025-11-10



#50781648提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
# 25353: 排队
# 题意: 每次只能交换相邻且身高差不超过 D 的人, 求字典序最小的最终排列
# 思路: 将允许交换关系看作有向无环图 (DAG), 每次取“入度为 0”的人中身高最小者输出。
# 实现: 多轮扫描—每轮找出当前所有入度为 0 的节点集合 S,
#       对 S 内按身高升序输出并从序列中删除, 重复直到序列为空。
# 注意: 最坏情况下复杂度为 O(N^2)。

import sys
input = sys.stdin.readline

def solve():
    line = input().split()
    if not line:
        return
    n, D = map(int, line)
    arr = [int(input()) for _ in range(n)]

    result = []
    cur = arr[:] # 当前剩余队列
```

基本信息

#: 50781648
题目: T25353
提交人:
25n2500011906(Little(25n2500011906))
内存: 21544kB
时间: 148ms
语言: Python3
提交时间: 2025-11-10 17:33:25



#50779594提交状态

查看 提交 统计 提问

状态: Time Limit Exceeded

源代码

```
n, d = list(map(int, input().split()))
js = []
for i in range(n):
    j = int(input())
    js.append([j, i])
# 输入队列中所有人的身高和初始状态的序号
for i in range(n):
    pas = 0
    # pas变量表示这位同学最多还能往前排的位数
    while i - 1 - pas >= 0 and abs(js[i - 1 - pas][0] - js[i][0]) <= d:
        pas += 1
    # 求出pas的值
    if pas >= 1:
        for t in range(i - pas, i):
            if js[t][0] > js[i][0]:
                js[i][1] = js[t][1]
                # 如果后面同学的身高较低就尽可能往前面站
                for g in range(t, i):
                    js[g][1] += 1
                # 队列先不动，只是改变序号
                break
js.sort(key=lambda x: x[1])
# 按照序号重新排队，此时就能得到身高的字典序最小的队列
for j in js:
    print(j[0])
# 从前往后依次报出每一位同学的身高
```

基本信息

#: 50779594
题目: T25353
提交人: 25n2500011906(Little(25n2500011906))
内存: 97820kB
时间: 8959ms
语言: PyPy3
提交时间: 2025-11-10 15:35:14

2. 学习总结和收获

这次拿到AC4，较上次有进步，但是解决难题的能力还有待提高。

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。

本周期中周，我除参加月考外还参加了化学安全、英语、高等数学的考试，额外练习就很少。

周日考完数学后回寝室参加了LeetCode的第475场周赛，明显比上周的那场难，只能做出第一题，第二题题面和第一题一样，但是数据更大，我就一直TLE，后两题也没有做出来，不是很理想。

第 475 场周赛 排名

已结束

全国

全球

大模型

1687 | 12 人 AK!

排名	用户名	得分	完成时间	题目1 (3)	题目2 (4)	题目3 (5)	题目4 (8)
1209	我	3	00:25:28	00:25:28	00:02:56	00:10:57	00:52:03 *
1	weiqizheng	20	00:57:03	00:02:32	00:02:56	00:10:57	00:56:57 *
2	TsReaper	20	01:01:57	00:01:25	00:03:49	00:10:29	00:56:57 *
3	w285714	20	01:05:57	00:02:00	00:01:47	00:18:20 * <td>00:55:57 *</td>	00:55:57 *
4	82年的muggle	20	01:14:13	00:02:32	00:02:51	00:12:38	01:04:13 *10min
5	lyongwolf	20	01:15:26	00:03:28	00:03:19	00:11:16	01:00:26 *15min
6	FReQuenter	20	01:18:15	00:02:48	00:06:08	00:14:56 *10min	00:53:15 *15min
7	龙猫	20	01:20:48	00:02:13 * <td>00:03:59</td> <td>00:11:34</td> <td>00:50:48 *25min</td>	00:03:59	00:11:34	00:50:48 *25min
8	Mkswll	20	01:21:29	00:02:42	00:02:36	00:10:08	00:56:29 *25min
9	LinkZelda	20	01:25:58	00:02:16	00:05:06	00:16:21 * <td>01:00:58 *20min</td>	01:00:58 *20min
10	Stressed Visves...	20	01:28:10	00:01:54	00:01:44	00:06:06	01:08:10 *20min
11	gooday	20	01:42:29	00:02:19	00:04:01	00:16:30	01:22:29 *