

Assignment #1: 摸底考试

Updated 11:15 GMT+8 Sep 20, 2023

2023 fall, Compiled by Xinjie Song, Phy

Markdown (用 <https://typoraio.cn> 编辑) 格式文件在, <https://github.com/GMyhf/2023fall-cs101>

课程号: 04831410 课程名: 计算概论(B)	班号: 12
上课时间: 1-16周 每周 周二 7-9节	地点: 理教208
上机时间: 2-15周 每周 周四 7-8节 期末机考时间: 第16周 周四 7-8节	地点: 理科1号楼计算中心, 二层楼的6号和三层楼的7号机房
助教: 张哲瑞、张以宁、彭亦男、涂程颖、陈威宇	在课程微信群中的名字是“TA-”开始, 地点: 理科1号楼1220

说明:

- 1) 为了尽量拉齐大家的学习基础, 尤其是督促零基础同学的强化学习, 第一周就进行了语法摸底考试。鉴于教学管理平台Canvas 9月22日开始使用, 这之前大家写好作业, 先自己保存, 之后通知提交到Canvas。
- 2) 摸底考试: AC5。摸底题目都在“练习”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。
- 3) 请把每个题目解题思路(可选), 源码Python, 或者C++/C (已经在Codeforces/Openjudge上AC), 截图(包含Accepted, 学号, 或者姓名+学号), 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。
- 4) 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”。提交后, 助教看到画面如下, 有同学清晰头像、提交文件有pdf、作业评论有md或者doc。

Assignment #3: Programming Practice
截止：2022年9月27日 23:59 - 04831410

212/212
已评分

3.92 / 4 (98%)
平均分

1/212

← 恒环学院 →

第 1 页, 共 2 页

282A. Bit++, implementation
<https://codeforces.com/problemset/problem/282/A>
t=int(input());c=0
while(t):
 strn=input()
 i=0
 while(i<len(strn)):
 if strn[i]=='+':
 c+=1;i+=2
 elif strn[i]=='-':
 c-=1;i+=2
 elif strn[i]=='X':
 c+=0;i+=1
 # c=count+c

 t-=1
 print(c)

已提交：2022年 9月26日 13:27

字数统计: 124 个字

提交的文件: (点击加载)
[Assignment 3.pdf](#)

测验
评分 共4
4

作业评论
[查看附件文件](#)
[Assignment 3.md](#)
恒城市与环境学院, 2022年 9月26日 13:28
[下载提交项评论](#)

5) 同学完成作业的时候, 就是这个模版文件中修改补充好。为便于助教批改作业, 请尽量不要删除其他文字。

6) 如果不能在截止前提交作业, 请写明原因。

编程环境

操作系统: Windows 11 22H2

Python编程环境: PyCharm 2023.2 (Community Edition)

C/C++编程环境: g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 必做题目

OJ02750: 鸡兔同笼

math, <http://cs101.openjudge.cn/practice/02750>

一个笼子里面关了鸡和兔子（鸡有2只脚，兔子有4只脚，没有例外）。已经知道了笼子里面脚的总数 a ，问笼子里面至少有多少只动物，至多有多少只动物。

输入

一行，一个正整数 a ($a < 32768$)。

输出

一行，包含两个正整数，第一个是最少的动物数，第二个是最多的动物数，两个正整数用一个空格分开。

如果没有满足要求的答案，则输出两个0，中间用一个空格分开。

样例输入

样例输出

```
5 10
```

【宋昕杰，物理学院，2023年秋】

思路：首先判断输入的情况是否存在，即如果a为奇数，直接输出<0 0>，否则计算最少的动物数（假设全部为兔子，若有剩余则为鸡）和最多的动物数（假设全部为鸡）并用f字符串输出。

Python3 代码

```
a = int(input())

if a % 2 != 0:
    print('0 0')
else:
    min_num = a // 4 + (a % 4) // 2
    max_num = a // 2
    print(f'{min_num} {max_num}')
```

Python代码运行截图



C++ 代码

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
```

```

if(a % 2 != 0)
{
    cout << "0 0";
}
else
{
    cout << int(a / 4) + (a % 4) / 2 << ' ' << a / 2;
}

return 0;
}

```

C++ 代码运行截图



OJ02733: 判断闰年

math, <http://cs101.openjudge.cn/practice/02733>

判断某年是否是闰年。

输入

输入只有一行，包含一个整数a (0 < a < 3000)

输出

一行，如果公元a年是闰年输出Y，否则输出N

样例输入

2006

样例输出

N

提示

公历纪年法中，能被4整除的大多是闰年，但能被100整除而不能被400整除的年份不是闰年，能被3200整除的也不是闰年，如1900年是平年，2000年是闰年，3200年不是闰年。

【宋昕杰，物理学院，2023年秋】

思路：读取输入的年份，依次判断即可；其中，为了避免嵌套判断语句块，增加了bool型变量yes_or_no记录是否为闰年。

Python3 代码

```
year = int(input())

yes_or_no = False
if year % 4 == 0:
    yes_or_no = True
    if year % 100 == 0 and year % 400 != 0:
        yes_or_no = False
    elif year % 3200 == 0:
        yes_or_no = False

if yes_or_no:
    print('Y')
else:
    print('N')
```

Python代码运行截图

The screenshot shows the OpenJudge web interface. At the top, there are browser tabs for 'OpenJudge - OpenJudge' and 'OpenJudge - 提交状态'. The address bar shows 'cs101.openjudge.cn/practice/solution/41191216/'. The main content area has a header for 'CS101 / 练习' with tabs for '题目', '排名', '状态', and '提问'. Below this, the problem title is '#41191216提交状态' with links for '查看', '提交', '统计', and '提问'. The status is 'Accepted'. On the left, under '源代码', the Python code is shown. On the right, under '基本信息', the submission details are listed.

源代码

```
year = int(input())
yes_or_no = False
if year % 4 == 0:
    yes_or_no = True
    if year % 100 == 0 and year % 400 != 0:
        yes_or_no = False
    elif year % 3200 == 0:
        yes_or_no = False

if yes_or_no:
    print('Y')
else:
    print('N')
```

基本信息

- #: 41191216
- 题目: 02733
- 提交人: 23n2300011524
- 内存: 3628kB
- 时间: 27ms
- 语言: Python3
- 提交时间: 2023-09-15 10:48:04

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

C++ 代码

```
#include<iostream>
using namespace std;
int main()
{
    int year;
    bool bYesOrNo = false;
    cin >> year;

    if(year % 4 == 0)
    {
        bYesOrNo = true;
        if(year % 100 == 0 and year % 400 != 0)
        {
            bYesOrNo = false;
        }
        else if(year % 3200 == 0)
        {
            bYesOrNo = false;
        }
    }
    if(bYesOrNo)
    {
        cout << 'Y';
    }
    else
    {
        cout << 'N';
    }
    return 0;
}
```

C++ 代码运行截图

OpenJudge - OpenJudge x OpenJudge - 提交状态 x +

不安全 | cs101.openjudge.cn/practice/solution/41191208/

OpenJudge 题目ID, 标题, 描述 23n2300011524 信箱 账号

CS101 / 练习

题目 排名 状态 提问

#41191208提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
#include<iostream>
using namespace std;
int main()
{
    int year;
    bool bYesOrNo = false;
    cin >> year;

    if(year % 4 == 0)
    {
        bYesOrNo = true;
        if(year % 100 == 0 and year % 400 != 0)
        {
            bYesOrNo = false;
        }
        else if(year % 3200 == 0)
        {
            bYesOrNo = false;
        }
    }
    if(bYesOrNo)
    {
        cout << 'Y';
    }
    else
    {
        cout << 'N';
    }
    return 0;
}
```

基本信息

#: 41191208
题目: 02733
提交人: 23n2300011524
内存: 248kB
时间: 7ms
语言: G++
提交时间: 2023-09-15 10:47:22

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

OJ01218:THE DRUNK JAILER

<http://cs101.openjudge.cn/practice/01218/>

A certain prison contains a long hall of n cells, each right next to each other. Each cell has a prisoner in it, and each cell is locked.

One night, the jailer gets bored and decides to play a game. For round 1 of the game, he takes a drink of whiskey, and then runs down the hall unlocking each cell. For round 2, he takes a drink of whiskey, and then runs down the hall locking every other cell (cells 2, 4, 6, ...). For round 3, he takes a drink of whiskey, and then runs down the hall. He visits every third cell (cells 3, 6, 9, ...). If the cell is locked, he unlocks it; if it is unlocked, he locks it. He repeats this for n rounds, takes a final drink, and passes out.

Some number of prisoners, possibly zero, realizes that their cells are unlocked and the jailer is incapacitated. They immediately escape.

Given the number of cells, determine how many prisoners escape jail.

输入

The first line of input contains a single positive integer. This is the number of lines that follow. Each of the following lines contains a single integer between 5 and 100, inclusive, which is the number of cells n .

输出

For each line, you must print out the number of prisoners that escape when the prison has n cells.

样例输入

```
2
5
100
```

样例输出

2

10

来源

Greater New York 2002

【宋昕杰，物理学院，2023年秋】

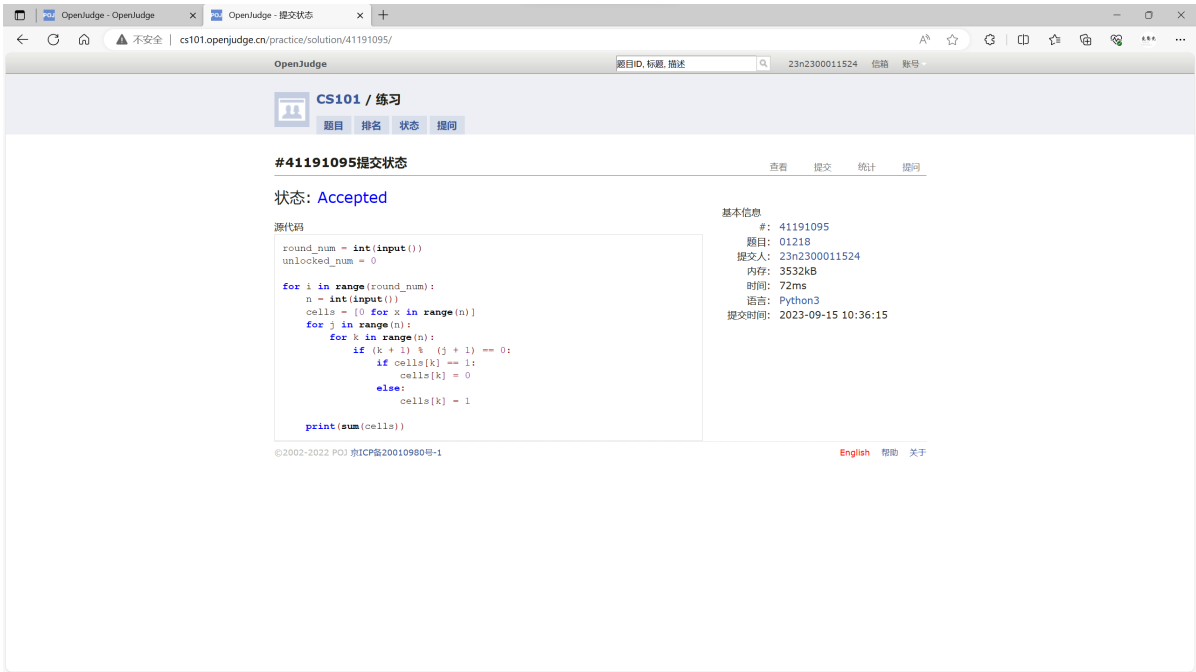
思路：生成一个列表cells用来记录每个隔间的状态，其中0代表锁定，1代表解锁，这样最后只需要输出sum(cells)即可；接下来进行n轮循环，模拟狱卒n次开锁；最后输出sum(cells)。

Python3 代码

```
round_num = int(input())
for i in range(round_num):
    n = int(input())
    cells = [0 for x in range(n)]
    for j in range(n):
        for k in range(n):
            if (k + 1) % (j + 1) == 0:
                if cells[k] == 1:
                    cells[k] = 0
            else:
                cells[k] = 1

    print(sum(cells))
```

Python代码运行截图



C++ 代码

```
#include<iostream>
using namespace std;
int main()
{
    int iRoundNumber;
    cin >> iRoundNumber;

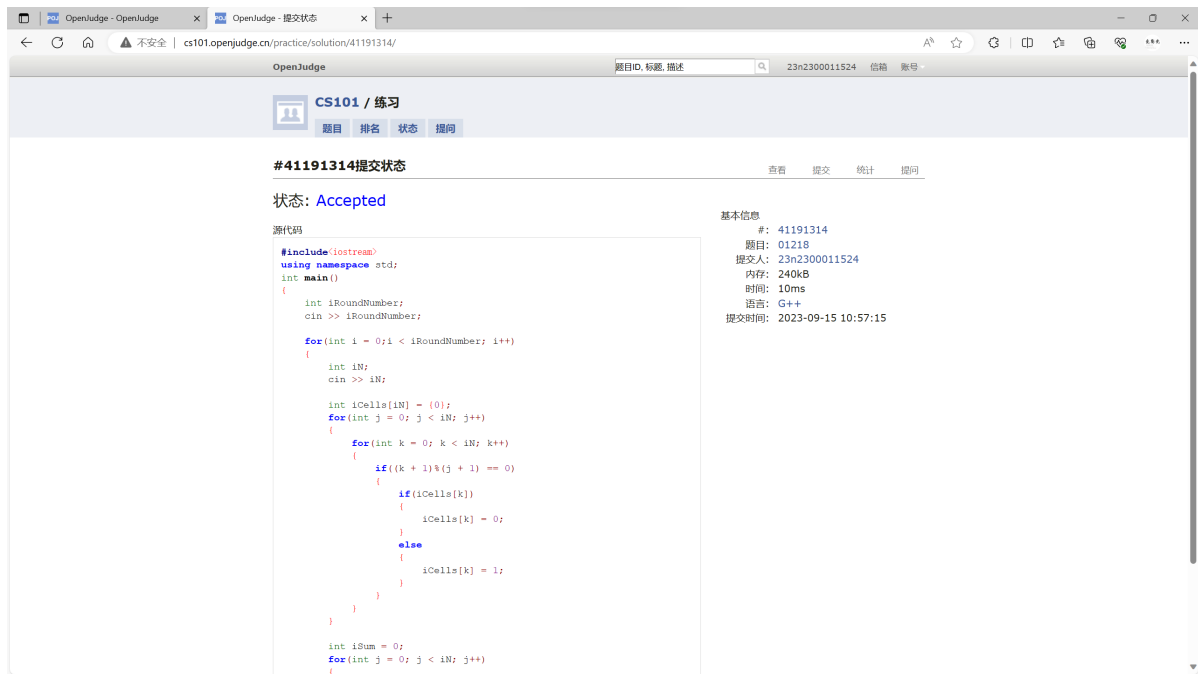
    for(int i = 0; i < iRoundNumber; i++)
    {
        int iN;
        cin >> iN;

        int icells[iN] = {0};
        for(int j = 0; j < iN; j++)
        {
            for(int k = 0; k < iN; k++)
            {
                if((k + 1)%(j + 1) == 0)
                {
                    if(icells[k])
                    {
                        icells[k] = 0;
                    }
                    else
                    {
                        icells[k] = 1;
                    }
                }
            }
        }

        int iSum = 0;
        for(int j = 0; j < iN; j++)
        {
            iSum += icells[j];
        }
        cout << iSum << endl;
    }

    return 0;
}
```

C++ 代码运行截图



OJ02689: 大小写字母互换

<http://cs101.openjudge.cn/practice/02689>

把一个字符串中所有出现的大写字母都替换成小写字母，同时把小写字母替换成大写字母。

输入

输入一行：待互换的字符串。

输出

输出一行：完成互换的字符串（字符串长度小于80）。

样例输入

```
If so, you already have a Google Account. You can sign in on the right.
```

样例输出

```
iF sO, yOu aLrEaDy hAvE a gOoGLe aCcOuNt. yOu cAn sIgN iN oN tHe rIgHt.
```

来源

计算概论05

【宋昕杰，物理学院，2023年秋】

思路：读取字符串，分别进行全部大写和全部小写操作，比对原字符串和操作后的字符串即可知道原来的字符是大写还是小写，并将所需字符追加到输出列表output中。

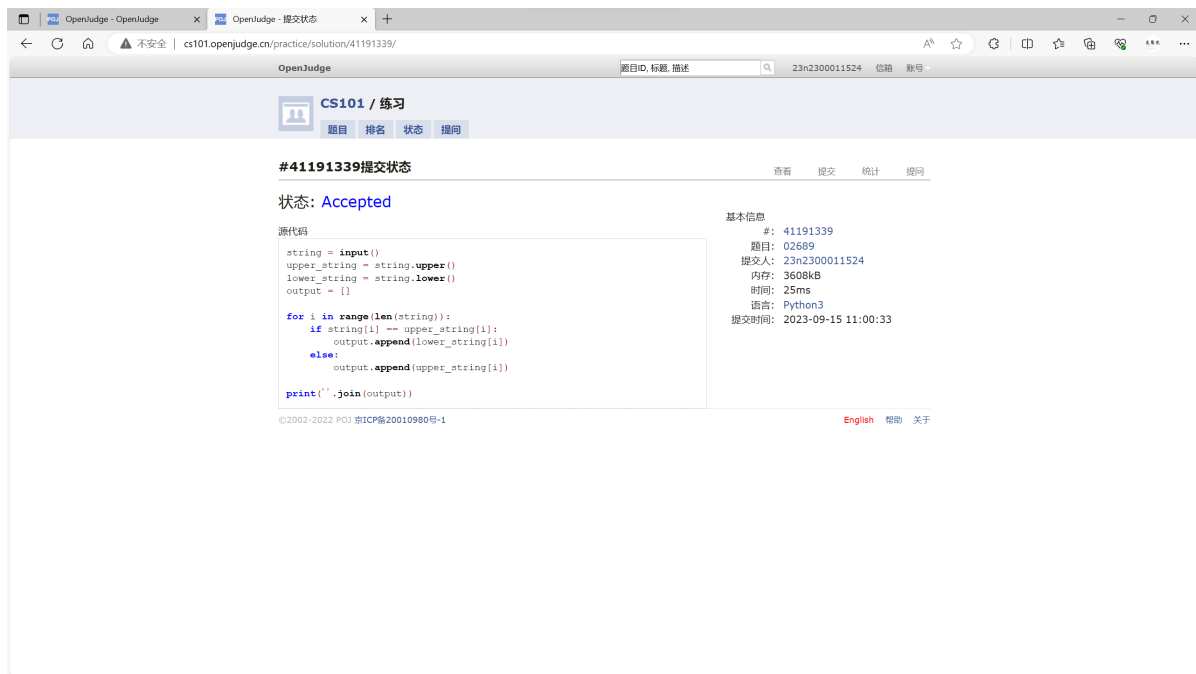
Python3 代码

```
string = input()
upper_string = string.upper()
lower_string = string.lower()
output = []

for i in range(len(string)):
    if string[i] == upper_string[i]:
        output.append(lower_string[i])
    else:
        output.append(upper_string[i])

print(''.join(output))
```

Python代码运行截图



OJ02808: 校门外的树

implementation, <http://cs101.openjudge.cn/practice/02808>

某校大门外长度为 L 的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在 L 的位置；数轴上的每个整数点，即0, 1, 2, ..., L ，都种有一棵树。

马路上有一些区域要用来建地铁，这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

输入

输入的第一行有两个整数 L ($1 \leq L \leq 10000$) 和 M ($1 \leq M \leq 100$)， L 代表马路的长度， M 代表区域的数目， L 和 M 之间用一个空格隔开。接下来的 M 行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点和终止点的坐标。

输出

输出包括一行，这一行只包含一个整数，表示马路上剩余的树的数目。

样例输入

```
500 3
150 300
100 200
470 471
```

样例输出

```
298
```

来源：noip2005普及组

【宋昕杰，物理学院，2023年秋】

思路：读取输入，生成列表road_list，其中1代表有树，0代表没有树，一次移去每个区间的树，最后输出sum(road_list)。

Python3 代码

```
l, m = map(int, input().split())
road_list = [1 for x in range(l + 1)]

for i in range(m):
    begin, end = map(int, input().split())
    for j in range(begin, end + 1):
        road_list[j] = 0

print(sum(road_list))
```

Python代码运行截图



OpenJudge - OpenJudge x OpenJudge - 提交状态 x +

cs101.openjudge.cn/practice/solution/41191488/

OpenJudge 题目ID, 标题, 描述 23n2300011524 信箱 账号

CS101 / 练习

题目 排名 状态 提问

#41191488提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
l, m = map(int, input().split())
road_list = [1 for x in range(l + 1)]

for i in range(m):
    begin, end = map(int, input().split())
    for j in range(begin, end + 1):
        road_list[j] = 0

print(sum(road_list))
```

基本信息

#: 41191488
题目: 02808
提交人: 23n2300011524
内存: 3652kB
时间: 52ms
语言: Python3
提交时间: 2023-09-15 11:17:55

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

C++ 代码

```
#include<iostream>
using namespace std;
int main()
{
    int iL, iM;

    cin >> iL >> iM;

    int iRoad[iL + 1];
    for(int i = 0; i < iL + 1; i++)
    {
        iRoad[i] = 1;
    }

    for(int i = 0; i < iM; i++)
    {
        int iBegin, iEnd;
        cin >> iBegin >> iEnd;

        for(int j = iBegin; j <= iEnd; j++)
        {
            iRoad[j] = 0;
        }
    }

    int iSum = 0;
    for(int i = 0; i < iL + 1; i++)
    {
        iSum += iRoad[i];
    }

    cout << iSum;
```

```
return 0;
}
```

C++ 代码运行截图



2. 选做题目

OJ25353: 排队

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

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
3
```

提示

【样例解释】

一种交换位置的过程如下：

```
7 7 3 6 2-> 7 7 6 3 2-> 7 7 6 2 3-> 7 6 7 2 3-> 6 7 7 2 3
```

【数据范围和约定】

对于 10% 的数据，满足 $N \leq 100$ ；

对于另外 20% 的数据，满足 $N \leq 5000$ ；

对于全部数据，满足 $1 \leq N \leq 10^5, 1 \leq D \leq 10^9, 1 \leq h_i \leq 10^9$ 。

【宋昕杰，物理学院，2023年秋】

思路：依次将最高的同学，第二高的同学.....向前移动直至不能再移动为止，但该算法必定超时，目前想不到更好的算法。

Python3 代码

```
n, d = map(int, input().split())
height_list = [int(input()) for x in range(n)]
sorted_height_list = sorted(height_list, reverse=True)
height_list.reverse()

for max_height in sorted_height_list:
    const_index = index = height_list.index(max_height)

    while index < n - 1:
        if abs(height_list[const_index] - height_list[index + 1]) > d:
            break
        index += 1
    height_list.insert(index + 1, height_list[const_index])
    del height_list[const_index]

height_list.reverse()
for height in height_list:
    print(height)
```

Python代码运行截图

OpenJudge - OpenJudge x OpenJudge - 提交状态 x +

不安全 | cs101.openjudge.cn/practice/solution/41189292/

OpenJudge 题目ID, 标题, 描述 23n2300011524 信箱 账号

CS101 / 练习

题目 排名 状态 提问

#41189292提交状态 查看 提交 统计 提问

状态: Time Limit Exceeded

源代码

```
n, d = map(int, input().split())
height_list = [int(input()) for x in range(n)]
sorted_height_list = sorted(height_list, reverse=True)
height_list.reverse()

for max_height in sorted_height_list:
    const_index = index = height_list.index(max_height)

    while index < n - 1:
        if abs(height_list[const_index] - height_list[index + 1]) > d:
            break
        index += 1
        height_list.insert(index + 1, height_list[const_index])
        del height_list[const_index]

height_list.reverse()
for height in height_list:
    print(height)
```

基本信息

#: 41189292
题目: 25353
提交人: 23n2300011524
内存: 8788kB
时间: 29942ms
语言: Python3
提交时间: 2023-09-14 22:31:43

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

C++ 代码

```
#include<iostream>
#include<algorithm>
#include<functional>
using namespace std;

int main()
{
    int iN, iD, iTemp;
    bool bNotDone = true;

    cin >> iN >> iD;
    int iHeight[iN], iHeightSorted[iN];

    for(int i = 0; i < iN; i++)
    {
        cin >> iHeight[i];
        iHeightSorted[i] = iHeight[i];
    }
    sort(iHeightSorted, iHeightSorted + iN, greater<int>());

    for(int i = 0; i < iN; i++)
    {
        int iIndex = 0;
        for(iIndex = iN - 1; iIndex >= 0; iIndex--)
        {
            if(iHeight[iIndex] == iHeightSorted[i])
            {
                break;
            }
        }

        for(; iIndex > 0; iIndex--)
        {
```



```

        if(abs(iHeight[iIndex] - iHeight[iIndex - 1]) <= iD)
        {
            iTemp = iHeight[iIndex];
            iHeight[iIndex] = iHeight[iIndex - 1];
            iHeight[iIndex - 1] = iTemp;
        }
        else
        {
            break;
        }
    }
}
for(int i = 0; i < iN; i++)
{
    cout << iHeight[i] <<endl;
}
}

```

C++ 代码运行截图



3. 学习总结和收获

考试中1-5题仅书写了Python代码，完成作业时补充了其中4道题目的C++代码，学习了一些C++中的语法，主要是C++中数据的读取方式；考试面对第6题时，使用Python提交后发现超时，尝试换用C++代码但效果甚微，考试后将C++代码中的高时间复杂度排序算法换成已有库中的快速排序函数，但仍然超时。

课下，按照闫老师自编教材上的例题顺序完成了相关题目，每日新增题目至少完成了其中1道相对简单的题目。

截止至2023-9-20 12:00，OJ完成题目40道，CF完成题目25道。

附录

如果设好了 typora的图床，md文件中图片在其他地方也能看见。因为图片存在云端。如果不好设置，注意导出的pdf文件包含图片。

Typora+PicGo+Github解决个人博客图片上传问题 <https://zhuanlan.zhihu.com/p/367529569>