Assignment #C: 矩阵、递归、贪心、和dfs simlar

Updated 1126 GMT+8 Nov 28, 2023

2023 fall, Complied by 同学的姓名、院系

说明：

本周作业还是难题较多，建议提前开始作业，如果耗时太长，直接找答案看。两个题解，经常更新。所以最好从这个链接下载最新的，https://github.com/GMyhf/2020fall-cs101 。

1）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, 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。

2）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、作业评论有md或者doc。

3）如果不能在截止前提交作业，请写明原因。

编程环境

（请改为同学的操作系统、编程环境等）

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

如果耗时太长，直接看解题思路，或者源码

CF1881C. Perfect Square

brute force, implementation, 1200, https://codeforces.com/problemset/problem/1881/C

黄源森推荐：”一个一般的矩阵“。感觉现在CF problemset第一页的题（难度1000+的）都不是那么好做。

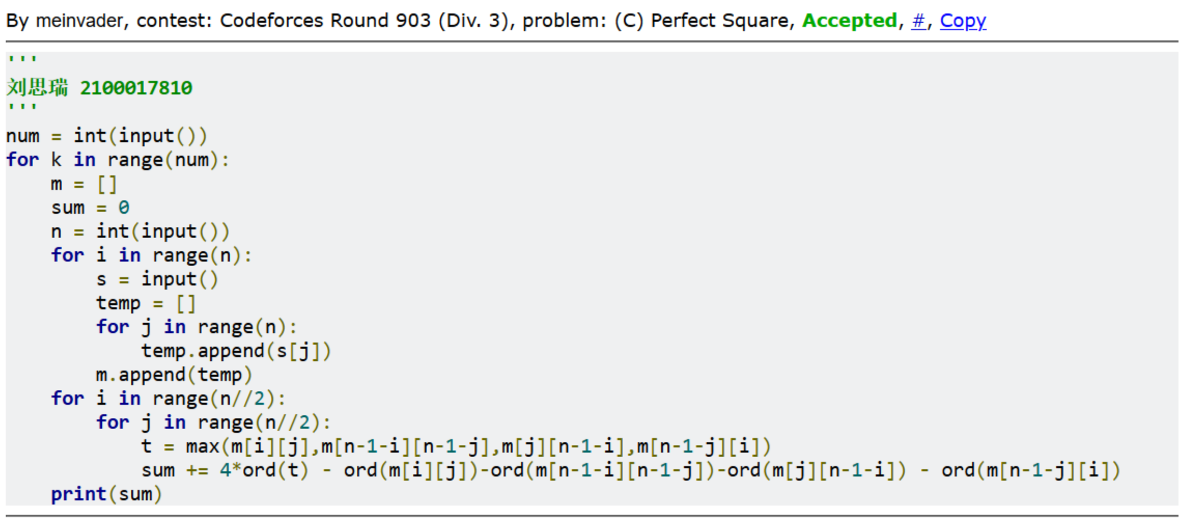
思路：

四个字符一组进行比对

代码

```python
'''
刘思瑞 2100017810
'''
num = int(input())
for k in range(num):
    m = []
    sum = 0
    n = int(input())
    for i in range(n):
        s = input()
        temp = []
        for j in range(n):
            temp.append(s[j])
        m.append(temp)
    for i in range(n//2):
        for j in range(n//2):
            t = max(m[i][j],m[n-1-i][n-1-j],m[j][n-1-i],m[n-1-j][i])
            sum += 4*ord(t) - ord(m[i][j])-ord(m[n-1-i][n-1-j])-ord(m[j][n-1-i]) - ord(m[n-1-j][i])
    print(sum)
```

代码运行截图



By meinvader, contest: Codeforces Round 903 (Div. 3), problem: (C) Perfect Square, **Accepted**, #, Copy

```python
'''
刘思瑞 2100017810
'''
num = int(input())
for k in range(num):
    m = []
    sum = 0
    n = int(input())
    for i in range(n):
        s = input()
        temp = []
        for j in range(n):
            temp.append(s[j])
        m.append(temp)
    for i in range(n//2):
        for j in range(n//2):
            t = max(m[i][j],m[n-1-i][n-1-j],m[j][n-1-i],m[n-1-j][i])
            sum += 4*ord(t) - ord(m[i][j])-ord(m[n-1-i][n-1-j])-ord(m[j][n-1-i]) - ord(m[n-1-j][i])
    print(sum)
```

OJ02694: 波兰表达式

recursion, data structure, http://cs101.openjudge.cn/practice/02694/

思路：

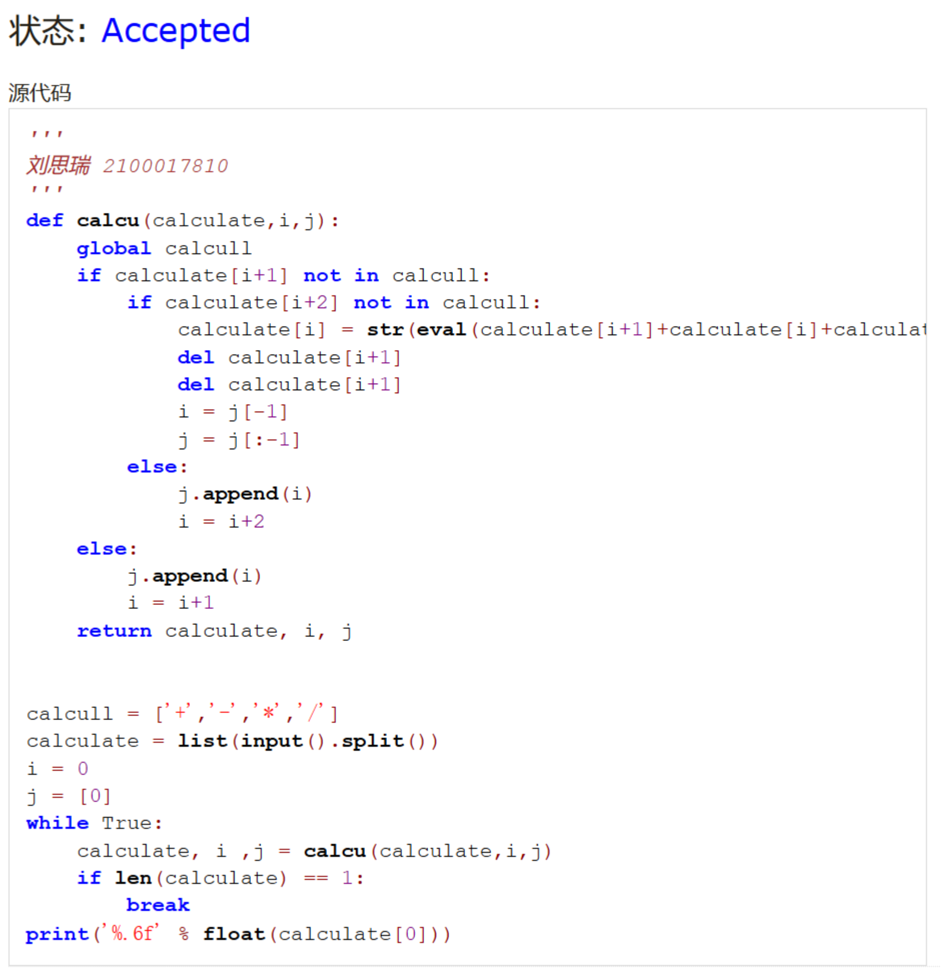
直接递归，但是学到了直接计算表达式的函数

代码

```python
'''
刘思瑞 2100017810
'''
def calcu(calculate,i,j):
    global calcull
    if calculate[i+1] not in calcull:
        if calculate[i+2] not in calcull:
            calculate[i] = 
str(eval(calculate[i+1]+calculate[i]+calculate[i+2]))
            del calculate[i+1]
            del calculate[i+1]
            i = j[-1]
            j = j[:-1]
        else:
            j.append(i)
            i = i+2
    else:
        j.append(i)
        i = i+1
    return calculate, i, j


calcull = ['+','-','*','/']
calculate = list(input().split())
i = 0
j = [0]
while True:
    calculate, i ,j = calcu(calculate,i,j)
    if len(calculate) == 1:
        break
print('%.6f' % float(calculate[0]))
```

代码运行截图



状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def calcu(calculate,i,j):
    global calcull
    if calculate[i+1] not in calcull:
        if calculate[i+2] not in calcull:
            calculate[i] = str(eval(calculate[i+1]+calculate[i]+calculat
            del calculate[i+1]
            del calculate[i+1]
            i = j[-1]
            j = j[:-1]
        else:
            j.append(i)
            i = i+2
    else:
        j.append(i)
        i = i+1
    return calculate, i, j


calcull = [’+’,’-’,’*’,’/’]
calculate = list(input().split())
i = 0
j = [0]
while True:
    calculate, i ,j = calcu(calculate,i,j)
    if len(calculate) == 1:
        break
print(’%.6f’ % float(calculate[0]))
```

OJ18160: 最大连通域面积

dfs similar, http://cs101.openjudge.cn/practice/18160

思路:

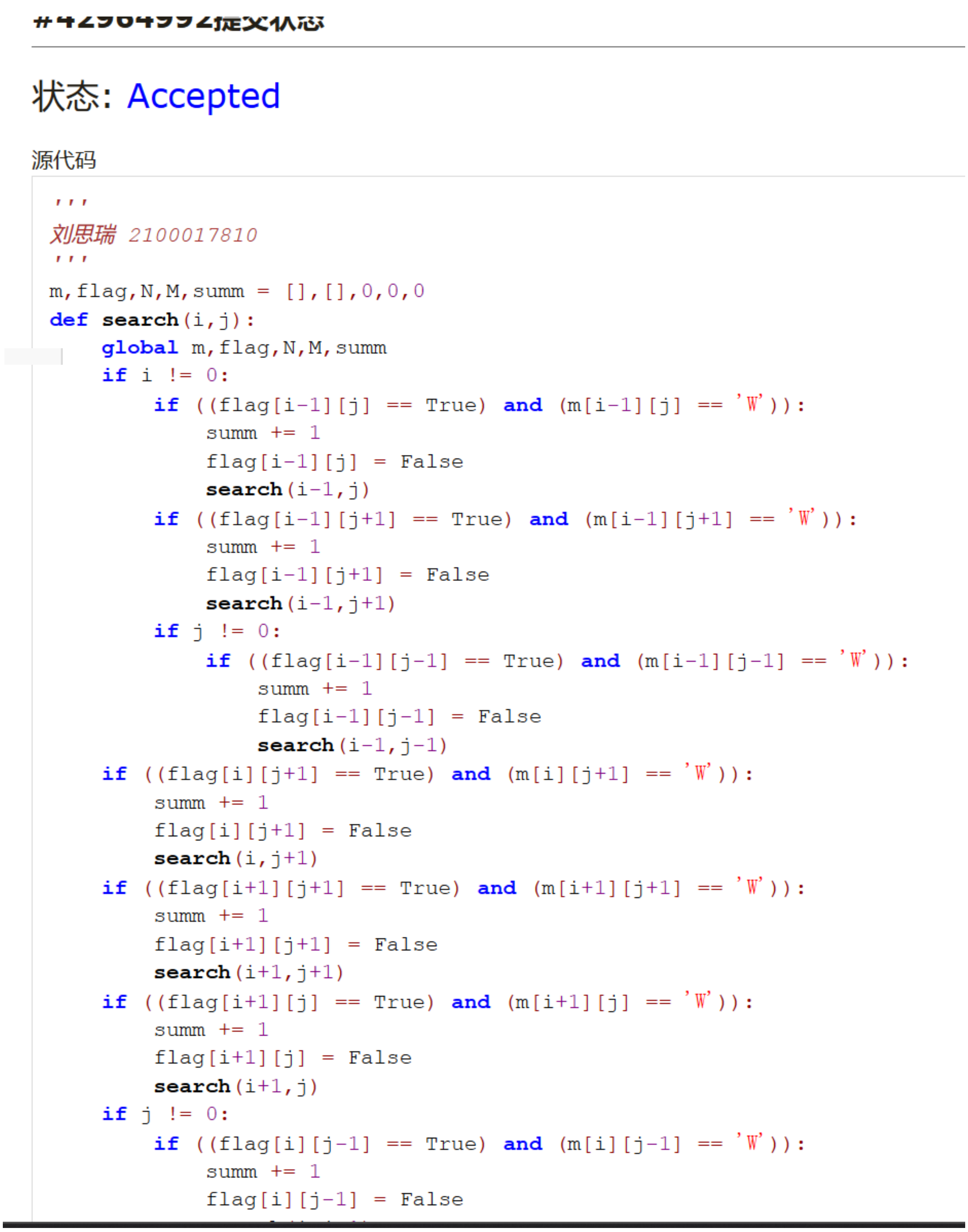
递归搜索

代码

```
'''
刘思瑞 2100017810
'''
m,flag,N,M,summ = [],[],0,0,0
def search(i,j):
    global m,flag,N,M,summ
    if i != 0:
        if ((flag[i-1][j] == True) and (m[i-1][j] == 'W')):
            summ += 1
            flag[i-1][j] = False
            search(i-1,j)
        if ((flag[i-1][j+1] == True) and (m[i-1][j+1] == 'W')):
            summ += 1
            flag[i-1][j+1] = False
            search(i-1,j+1)
        if j != 0:
            if ((flag[i-1][j-1] == True) and (m[i-1][j-1] == 'W')):
                summ += 1
                flag[i-1][j-1] = False
                search(i-1,j-1)
    if ((flag[i][j+1] == True) and (m[i][j+1] == 'W')):
        summ += 1
        flag[i][j+1] = False
        search(i,j+1)
    if ((flag[i+1][j+1] == True) and (m[i+1][j+1] == 'W')):
        summ += 1
        flag[i+1][j+1] = False
        search(i+1,j+1)
    if ((flag[i+1][j] == True) and (m[i+1][j] == 'W')):
        summ += 1
        flag[i+1][j] = False
        search(i+1,j)
    if j != 0:
        if ((flag[i][j-1] == True) and (m[i][j-1] == 'W')):
            summ += 1
            flag[i][j-1] = False
            search(i,j-1)
        if ((flag[i+1][j-1] == True) and (m[i+1][j-1] == 'W')):
            summ += 1
            flag[i+1][j-1] = False
            search(i+1,j-1)
    return


num = int(input())
for k in range(num):
    m = []
    flag = []
    sum = 0
    N,M = map(int,input().split())
    for i in range(N):
        flag.append([True]*(M)+[False])
        s = input()
        temp = []
        for j in range(M):
            temp.append(s[j])
        temp.append('.')
        m.append(temp)
    m.append(['.']*(M+1))
    flag.append([False]*(M+1))
    for i in range(N):
        for j in range(M):
            if m[i][j] =='W' and flag[i][j] == True:
                summ = 1
                flag[i][j] = False
                search(i,j)
                sum = max(sum,summ)
    print(sum)
```

代码运行截图



#42964992提交状态

状态: Accepted

源代码

```python
'''
刘思瑞 2100017810
'''
m,flag,N,M,summ = [],[],0,0,0
def search(i,j):
    global m,flag,N,M,summ
    if i != 0:
        if ((flag[i-1][j] == True) and (m[i-1][j] == 'W')):
            summ += 1
            flag[i-1][j] = False
            search(i-1,j)
        if ((flag[i-1][j+1] == True) and (m[i-1][j+1] == 'W')):
            summ += 1
            flag[i-1][j+1] = False
            search(i-1,j+1)
        if j != 0:
            if ((flag[i-1][j-1] == True) and (m[i-1][j-1] == 'W')):
                summ += 1
                flag[i-1][j-1] = False
                search(i-1,j-1)
    if ((flag[i][j+1] == True) and (m[i][j+1] == 'W')):
        summ += 1
        flag[i][j+1] = False
        search(i,j+1)
    if ((flag[i+1][j+1] == True) and (m[i+1][j+1] == 'W')):
        summ += 1
        flag[i+1][j+1] = False
        search(i+1,j+1)
    if ((flag[i+1][j] == True) and (m[i+1][j] == 'W')):
        summ += 1
        flag[i+1][j] = False
        search(i+1,j)
    if j != 0:
        if ((flag[i][j-1] == True) and (m[i][j-1] == 'W')):
            summ += 1
            flag[i][j-1] = False
```

OJ02754: 八皇后

dfs, http://cs101.openjudge.cn/practice/02754

思路:

其实可以遍历到指定的元素最大值就停止

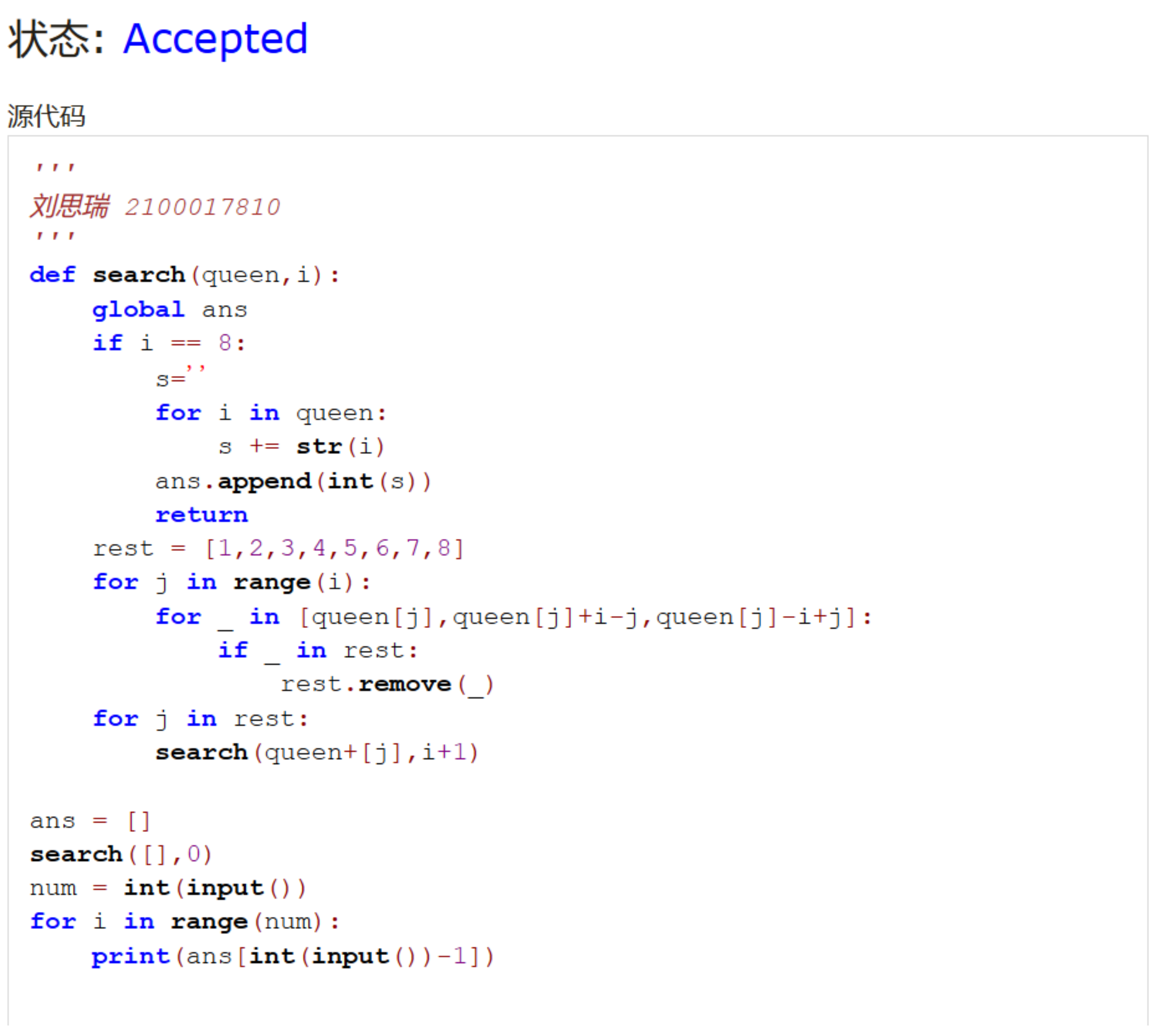
 不过有点太麻烦了感觉没必要就不写了

代码

```python
'''
刘思瑞 2100017810
'''
def search(queen,i):
    global ans
    if i == 8:
        s=''
        for i in queen:
            s += str(i)
        ans.append(int(s))
        return
    rest = [1,2,3,4,5,6,7,8]
    for j in range(i):
        for _ in [queen[j],queen[j]+i-j,queen[j]-i+j]:
            if _ in rest:
                rest.remove(_)
    for j in rest:
        search(queen+[j],i+1)

ans = []
search([],0)
num = int(input())
for i in range(num):
    print(ans[int(input())-1])
```

代码运行截图


状态: Accepted

源代码

```python
'''
刘思瑞 2100017810
'''
def search(queen,i):
    global ans
    if i == 8:
        s=''
        for i in queen:
            s += str(i)
        ans.append(int(s))
        return
    rest = [1,2,3,4,5,6,7,8]
    for j in range(i):
        for _ in [queen[j],queen[j]+i-j,queen[j]-i+j]:
            if _ in rest:
                rest.remove(_)
    for j in rest:
        search(queen+[j],i+1)

ans = []
search([],0)
num = int(input())
for i in range(num):
    print(ans[int(input())-1])
```

OJ18146: 乌鸦坐飞机

http://cs101.openjudge.cn/routine/18146/

查达闻推荐：乌鸦坐飞机和装箱子那道题很像，其实难度不比装箱子高 但是考虑的情况确实不少。

思路：

确实是贪心但是要考虑的太多了，大概就分为三种箱子，一种是4，一种是2，还有一种是4里面装了2变成1，在这些情况中有时候是要考虑用空间换类别的因此会非常麻烦

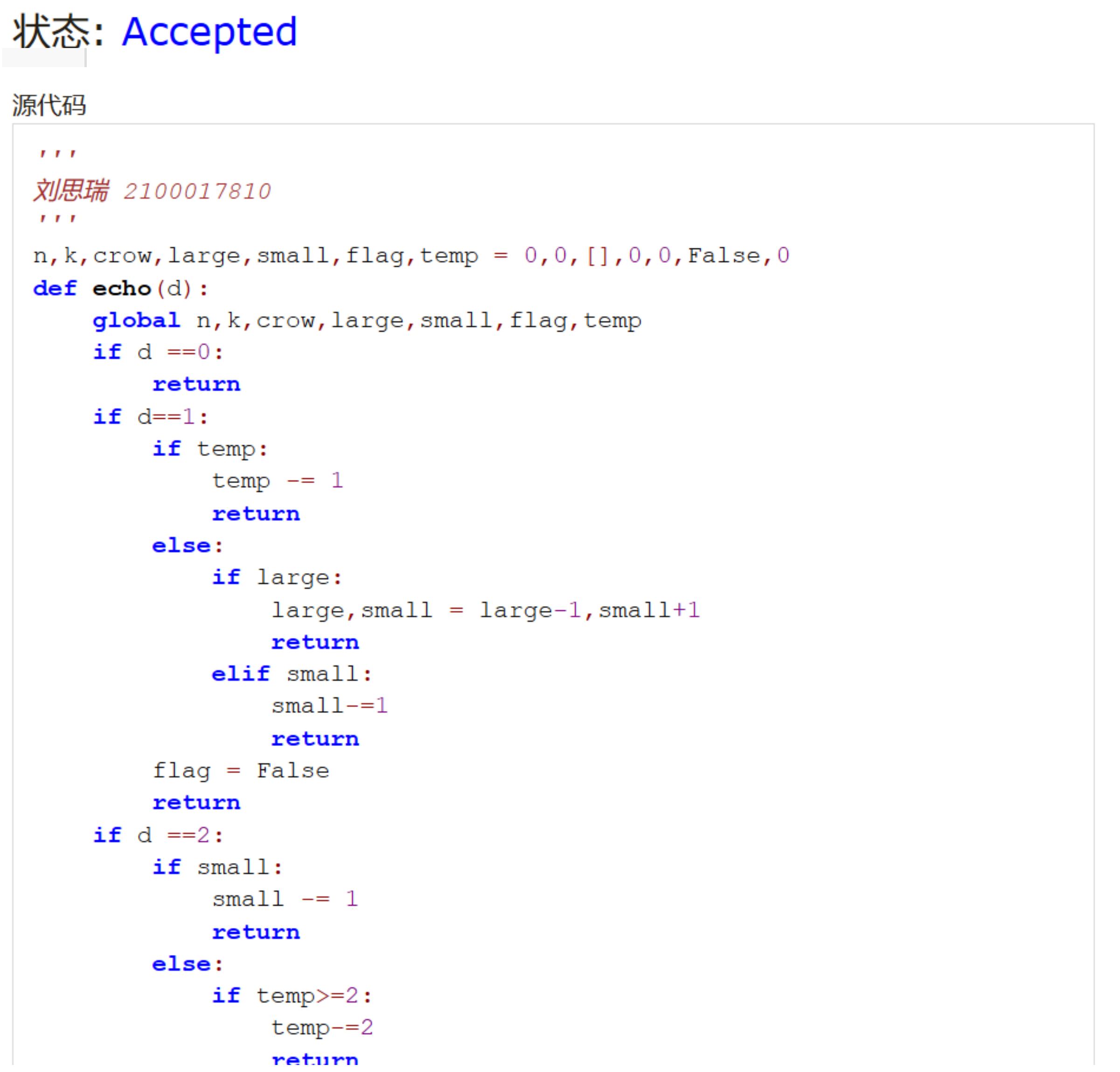
代码

```python
'''
刘思瑞 2100017810
'''
n,k,crow,large,small,flag,temp = 0,0,[],0,0,False,0
def echo(d):
    global n,k,crow,large,small,flag,temp
    if d ==0:
        return
    if d==1:
        if temp:
            temp -= 1
            return
        else:
            if large:
                large,small = large-1,small+1
                return
            elif small:
                small-=1
                return
        flag = False
        return
    if d ==2:
        if small:
            small -= 1
            return
        else:
            if temp>=2:
                temp-=2
                return
            if large:
                if temp:
                    large -=1
                    small+=1
                    temp -=1
                    return
                else:
                    large,temp = large-1,temp+1
                    return
            if temp>=2:
                temp-=2
                return
        flag = False
        return
    if d ==3:
        if large:
            large -= 1
            return
        else:
            if temp:
                if small:
                    temp -=1
                    small-=1
                    return
            else:
                if small:
                    if large:
                        small-=1
                        large-=1
                        temp+=1
                    elif small>=2:
                        small-=2
                        return
        flag = False
        return
    if d == 4:
        if large:
            large -= 1
            return
        else:
            if small>=2:
                small-=2
                return
            else:
                if small:
                    if temp>=2:
                        small-=1
                        temp-=2
                        return
                elif temp>=4:
                    temp-=4
                    return
        flag = False
        return

def define(c):
    global n,k,crow,large,small,flag,temp
    for i in range(k):
        rest = crow[i]//4
        for j in range(rest):
            echo(4)
        echo(crow[i]%4)
        if not flag:
            return 'NO'
    return 'YES'

n,k = map(int,input().split())
crow = list(map(int,input().split()))
small = 2*n
large = n
temp = 0
flag = True
crow.sort()
print(define(1))
```

代码运行截图


状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
n,k,crow,large,small,flag,temp = 0,0,[],0,0,False,0
def echo(d):
    global n,k,crow,large,small,flag,temp
    if d ==0:
        return
    if d==1:
        if temp:
            temp -= 1
            return
        else:
            if large:
                large,small = large-1,small+1
                return
            elif small:
                small-=1
                return
        flag = False
        return
    if d ==2:
        if small:
            small -= 1
            return
        else:
            if temp>=2:
                temp-=2
                return
```

OJ02287: 田忌赛马

greedy, http://cs101.openjudge.cn/practice/02287

思路:

按小马来比较

代码

```
'''
刘思瑞 2100017810
'''
def money(tian,king,num):
    tian.sort(reverse=True)
    king.sort(reverse=True)
    sum = 0
    while True:
        if tian == []:
            print(sum*200)
            return
        besttian = tian[0]
        worsetian = tian[-1]
        bestking = king[0]
        worseking = king[-1]
        if worsetian > worseking:
            tian,king = tian[:-1],king[:-1]
            sum+=1
            continue
        if worsetian < worseking:
            tian,king = tian[:-1],king[1:]
            sum-=1
            continue
        if worseking == worsetian:
            if bestking < besttian:
                tian,king = tian[:-1],king[:-1]
            elif bestking > besttian:
                tian,king = tian[:-1],king[1:]
                sum-=1
            else:
                if worsetian < bestking:
                    sum-=1
                tian,king = tian[:-1],king[1:]
            continue

while True:
    n = int(input())
    if not n:
        break
    tian = list(map(int,input().split()))
    king = list(map(int,input().split()))
    money(tian,king,n)
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def money(tian,king,num):
    tian.sort(reverse=True)
    king.sort(reverse=True)
    sum = 0
    while True:
        if tian == []:
            print(sum*200)
            return
        besttian = tian[0]
        worsetian = tian[-1]
        bestking = king[0]
        worseking = king[-1]
        if worsetian > worseking:
            tian,king = tian[:-1],king[:-1]
            sum+=1
            continue
        if worsetian < worseking:
            tian,king = tian[:-1],king[1:]
            sum-=1
            continue
        if worseking == worsetian:
            if bestking < besttian:
                tian,king = tian[:-1],king[:-1]
            elif bestking > besttian:
                tian,king = tian[:-1],king[1:]
                sum-=1
```

com...

2. 学习总结和收获

感觉这次作业主要是debug非常痛苦，特别是乌鸦坐飞机，实在是对着参考数据改的