# Assignment #F: 十全十美

Updated 1305 GMT+8 Dec 19, 2023

2023 fall, Complied by Xinjiesong, Phy

**说明：**

本周作业对零基础同学偏难，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**

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. 题目

如果耗时太长，直接看解题思路，或者源码

## 18155: 组合乘积

dfs, brute force, http://cs101.openjudge.cn/practice/18155

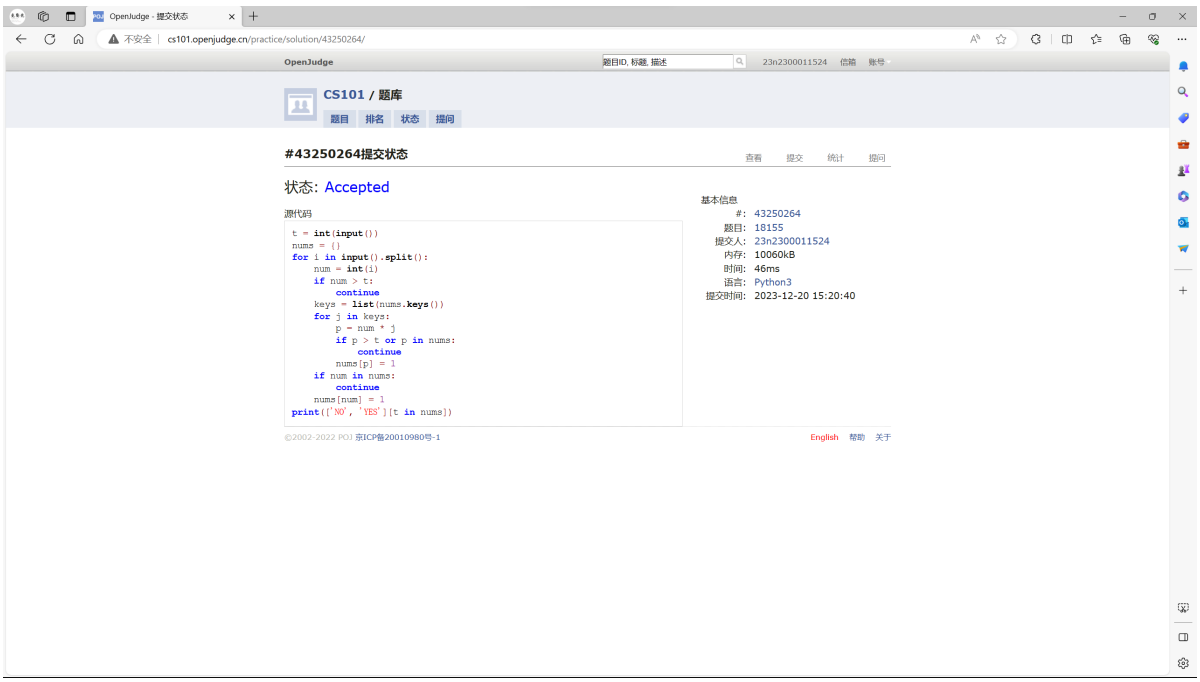
思路：桶门!

**代码**

```
t = int(input())
nums = {}
for i in input().split():
    num = int(i)
    if num > t:
        continue
    keys = list(nums.keys())
```

```
    for j in keys:
        p = num * j
        if p > t or p in nums:
            continue
        nums[p] = 1
    if num in nums:
        continue
    nums[num] = 1
print(['NO', 'YES'][t in nums])
```

代码运行截图




## 20106: 走山路

bfs, http://cs101.openjudge.cn/practice/20106/

思路：一开始写了常规BFS，无论如何优化，总是TLE；周五参考了胡同学的思路，写了拯救行动；再想起这道题，思路几乎一模一样，于是将队列改为堆，顺利AC

### 代码

```
import heapq

m, n, p = map(int, input().split())
t = [['#']*(n + 2)]
matrix = t + [['#'] + input().split() + ['#'] for _ in range(m)] + t[:]
for i in range(1, m + 1):
    for j in range(1, n + 1):
        if matrix[i][j] != '#':
            matrix[i][j] = int(matrix[i][j])
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
```

```
    x1 += 1
    y1 += 1
    x2 += 1
    y2 += 1
    if matrix[x1][y1] == '#' or matrix[x2][y2] == '#':
        print('NO')
        continue
    heap = []
    heapq.heapify(heap)
    visited = [[0] * (n + 2) for _ in range(m + 2)]
    heapq.heappush(heap, (0, x1, y1))
    min_cost = float('inf')
    while len(heap):
        c, x, y = heapq.heappop(heap)
        if visited[x][y]:
            continue
        t, visited[x][y] = matrix[x][y], 1
        if (x, y) == (x2, y2):
            print(c)
            break
        for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
            if matrix[x + dx][y + dy] == '#' or visited[x + dx][y + dy]:
                continue
            heapq.heappush(heap, (c + abs(matrix[x + dx][y + dy] - t), x + dx, y
+ dy))
    if not visited[x2][y2]:
        print('NO')
```

代码运行截图

# 27314: 一键换词

implementation, string, http://cs101.openjudge.cn/practice/27314/

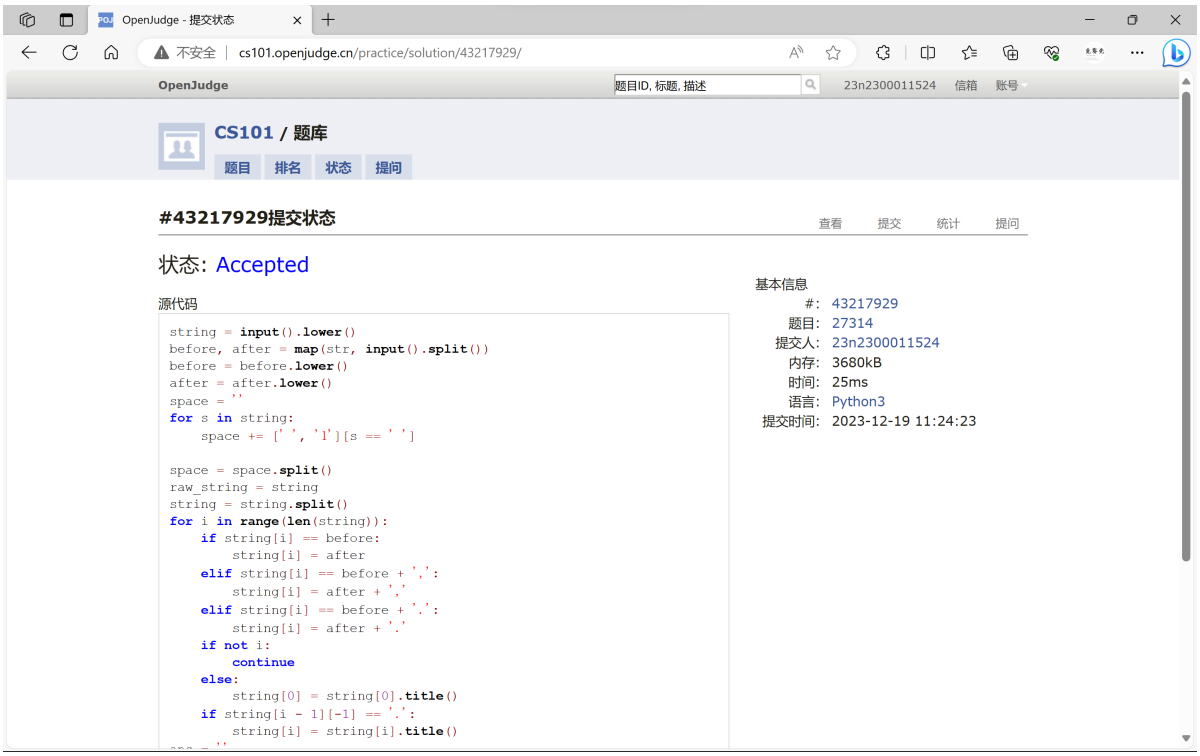
思路：先按顺序将输入内容拆为空格和字符，然后将字符中的目标词语替换；如果上一个词末尾有句号，现在这个词首字母大写；最后，将空格和替换完成的字符合并，输出即可

**代码**

```
string = input().lower()
before, after = map(str, input().split())
before = before.lower()
after = after.lower()
space = ''
for s in string:
    space += [' ', '1'][s == ' ']

space = space.split()
raw_string = string
string = string.split()
for i in range(len(string)):
    if string[i] == before:
        string[i] = after
    elif string[i] == before + ',':
        string[i] = after + ','
    elif string[i] == before + '.':
        string[i] = after + '.'
    if not i:
        continue
    else:
        string[0] = string[0].title()
    if string[i - 1][-1] == '.':
        string[i] = string[i].title()
ans = ''
if raw_string[0] == ' ':
    for i in range(len(space)):
        ans += ' '*len(space[i]) + string[i]
else:
    for i in range(len(space)):
        ans += string[i] + ' '*len(space[i])
if raw_string[-1] == ' ':
    ans += ' '*len(space[-1])
else:
    ans += string[-1]
print(ans)
```

代码运行截图




# 19961: 最大点数(外太空2048)

matrices, http://cs101.openjudge.cn/practice/19961/

思路：DFS，仅使用左移配合旋转即可节省实现游戏规则的代码量

### 代码

```
from copy import deepcopy

m, n, p = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(m)]
ans = set()


def left(ma):
    m, n = len(ma), len(ma[0])
    while True:
        c = False
        for i in range(m):
            for j in range(n - 1, 0, -1):
                if ma[i][j] and not ma[i][j - 1]:
                    ma[i][j], ma[i][j - 1] = 0, ma[i][j]
                    c = True
        if not c:
            break
    for i in range(m):
        for j in range(n - 1, 0, -1):
            if ma[i][j] == ma[i][j - 1]:
                ma[i][j - 1] *= 2
```

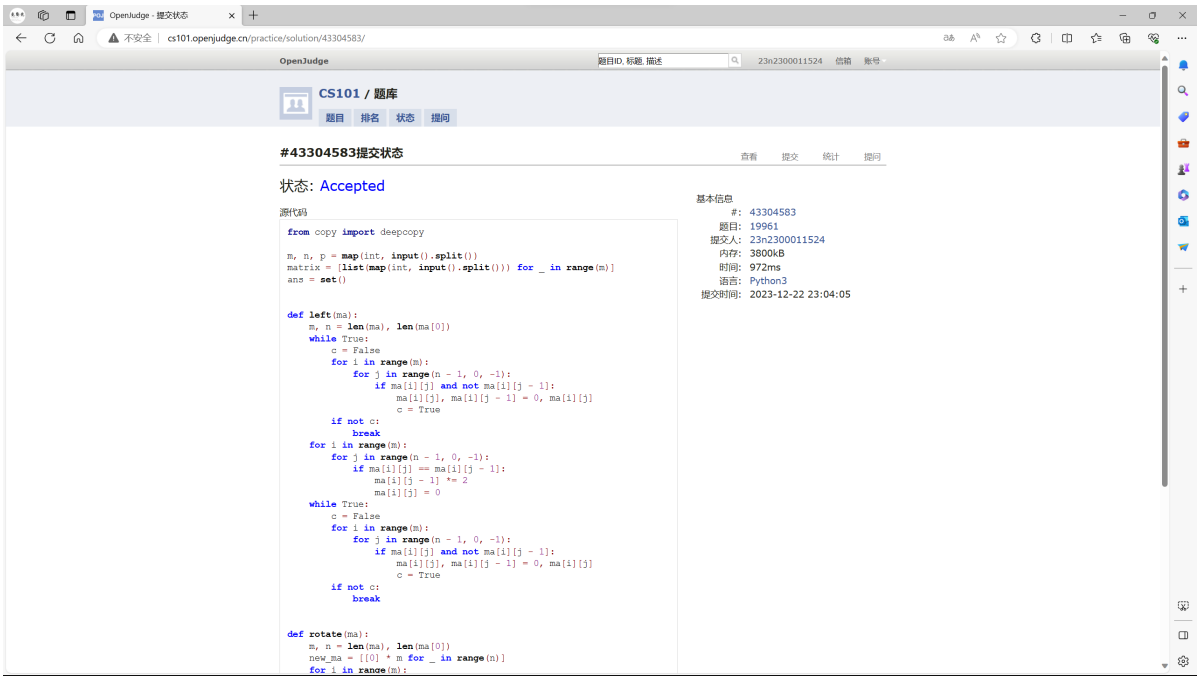
```
                ma[i][j] = 0
    while True:
        c = False
        for i in range(m):
            for j in range(n - 1, 0, -1):
                if ma[i][j] and not ma[i][j - 1]:
                    ma[i][j], ma[i][j - 1] = 0, ma[i][j]
                    c = True
        if not c:
            break


def rotate(ma):
    m, n = len(ma), len(ma[0])
    new_ma = [[0] * m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            new_ma[j][m - i - 1] = ma[i][j]
    return new_ma


def dfs(matrix, step):
    if not step:
        ans.add(max([max(i) for i in matrix]))
        return
    step -= 1
    for i in range(4):
        new_matrix = deepcopy(matrix)
        left(new_matrix)
        dfs(new_matrix, step)
        matrix = rotate(matrix)


dfs(matrix, p)
print(max(ans))
```

代码运行截图




# 27401: 最佳凑单

dp, sparse bucket, http://cs101.openjudge.cn/practice/27401/

思路：桶门!

### 代码

```
n, t = map(int, input().split())
prices = sorted(list(map(int, input().split())))
total = sum(prices)
if total < t:
    print(0)
    exit()

dic = {i: 0 for i in range(t)}
l = dic[0] = 1
keys = [0]
min_t = total
for i in range(n):
    for j in range(l):
        p = keys[j] + prices[i]
        if p > t:
            min_t = min(min_t, p)
            continue
        elif p == t:
            print(t)
            exit()
        if dic[p] == 0:
            dic[p] += 1
            l += 1
            keys.append(p)
```

```
print(min_t)
```

代码运行截图


# 27384: 候选人追踪

heap, http://cs101.openjudge.cn/practice/27384/

熊江凯，这题应该不超纲的，感觉还是挺好的

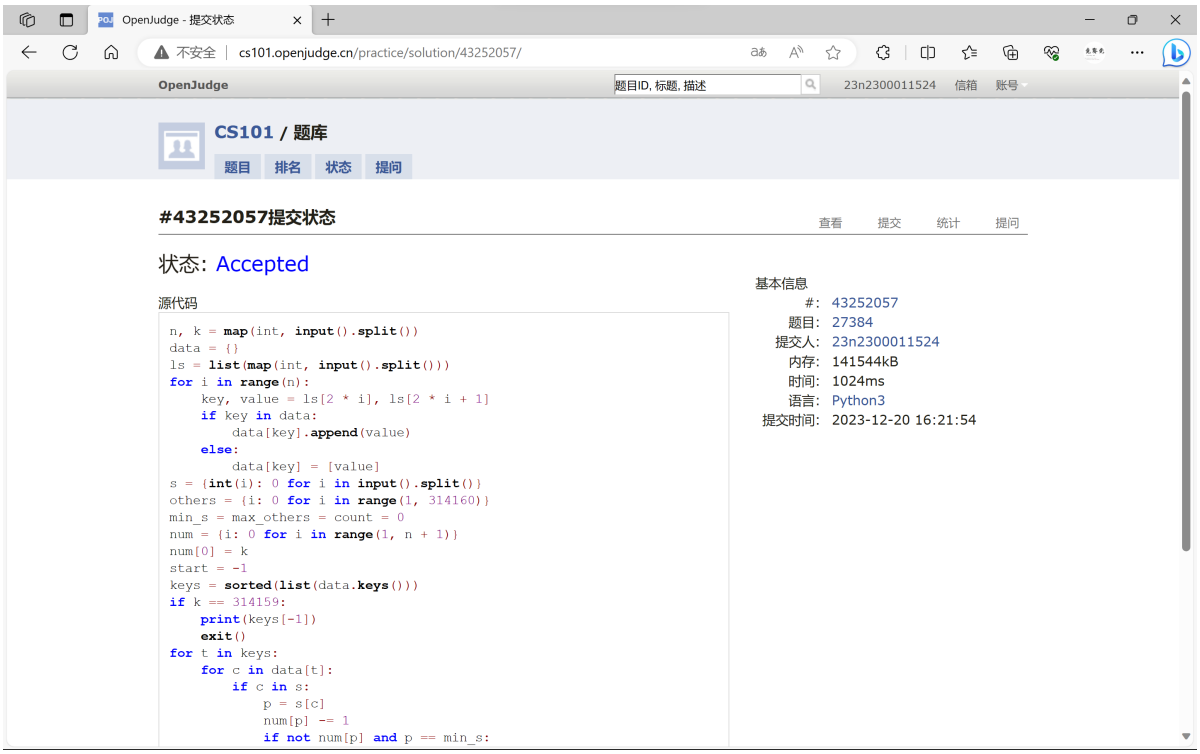
思路：桶门！但$k = 314159$太恶劣了

### 代码

```
n, k = map(int, input().split())
data = {}
ls = list(map(int, input().split()))
for i in range(n):
    key, value = ls[2 * i], ls[2 * i + 1]
    if key in data:
        data[key].append(value)
    else:
        data[key] = [value]
s = {int(i): 0 for i in input().split()}
others = {i: 0 for i in range(1, 314160)}
min_s = max_others = count = 0
num = {i: 0 for i in range(1, n + 1)}
num[0] = k
start = -1
```

```
keys = sorted(list(data.keys()))
if k == 314159:
    print(keys[-1])
    exit()
for t in keys:
    for c in data[t]:
        if c in s:
            p = s[c]
            num[p] -= 1
            if not num[p] and p == min_s:
                min_s += 1
            s[c] += 1
            num[p + 1] += 1
        else:
            others[c] += 1
            max_others = max(max_others, others[c])
    if max_others < min_s:
        if start == -1:
            start = t
    elif start != -1:
        count += t - start
        start = -1
print(count + [keys[-1] - start, 0][start == -1])
```

代码运行截图



# CF1883D. In Love

data structure, greedy, 1500, https://codeforces.com/problemset/problem/1883/D

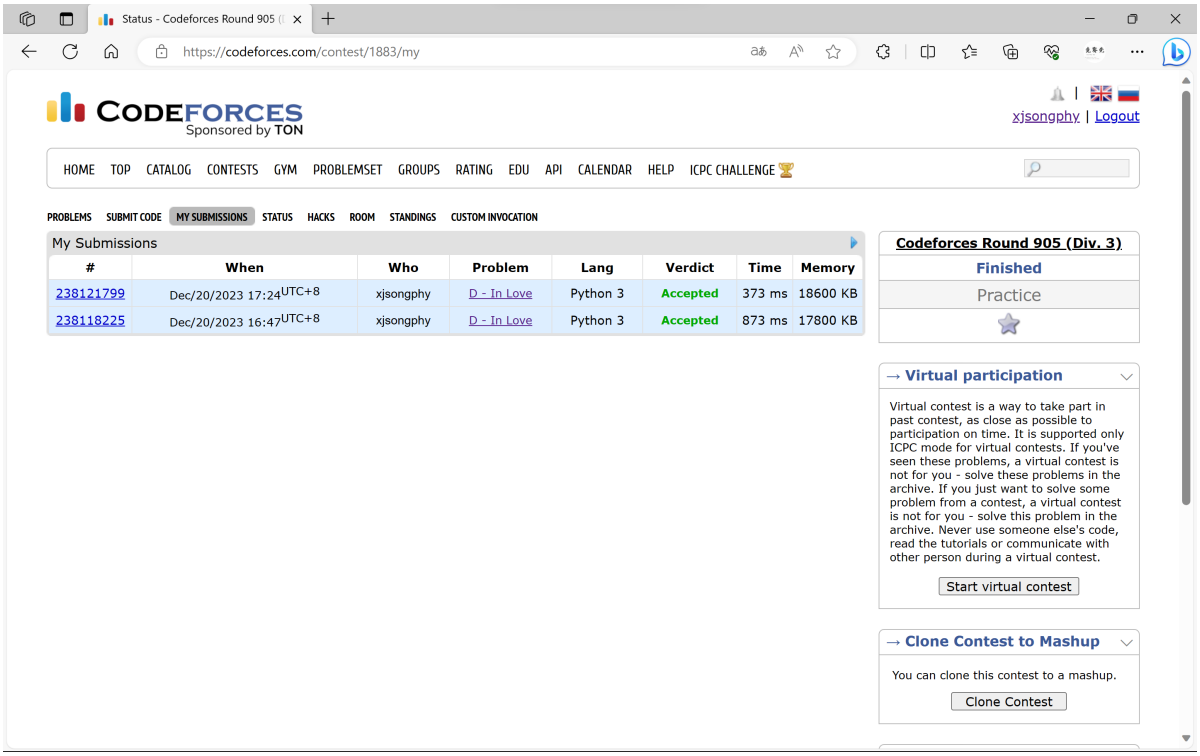
黄源森、查达闻推荐

思路：类似滑动窗口，减少求最大值最小值的次数。统一输出答案！统一输出答案！统一输出答案！单个输出答案耗时873ms，统一输出答案耗时373ms。

### 代码

```
ls, rs = {}, {}
max_l = min_r = -1
ans = ''
for i in range(int(input())):
    t, l, r = map(str, input().split())
    l, r = int(l), int(r)
    if t == '+':
        if l in ls:
            ls[l] += 1
        else:
            ls[l] = 1
        if r in rs:
            rs[r] += 1
        else:
            rs[r] = 1
        if max_l == -1:
            max_l = l
        else:
            max_l = max(max_l, l)
        if min_r == -1:
            min_r = r
        else:
            min_r = min(min_r, r)
    else:
        ls[l] -= 1
        rs[r] -= 1
        if rs[r] == 0:
            del rs[r]
            if r == min_r:
                if rs:
                    min_r = min(rs.keys())
                else:
                    max_l = min_r = -1
        if ls[l] == 0:
            del ls[l]
            if l == max_l and ls:
                max_l = max(ls.keys())
    ans += ['NO', 'YES'][max_l > min_r] + '\n'
print(ans.rstrip('\n'))
```

代码运行截图




# 2. 学习总结和收获

这次作业有些题目不看题解很难想，如果深刻贯彻学习落实了桶的思想，其余题目难度适中，也没有特别难。后面要抓紧查漏补缺了。

截至2023年12月22日，OJ完成题目167道，CF完成题目53道。