Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Complied by ==同学的姓名、院系==

说明：

1）The complete process to learn DSA from scratch can be broken into 4 parts:

- Learn about Time and Space complexities
- Learn the basics of individual Data Structures
- Learn the basics of Algorithms
- Practice Problems on DSA

2）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），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。

3）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"区有上传的md或者doc附件。

4）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

02945: 拦截导弹

http://cs101.openjudge.cn/practice/02945/

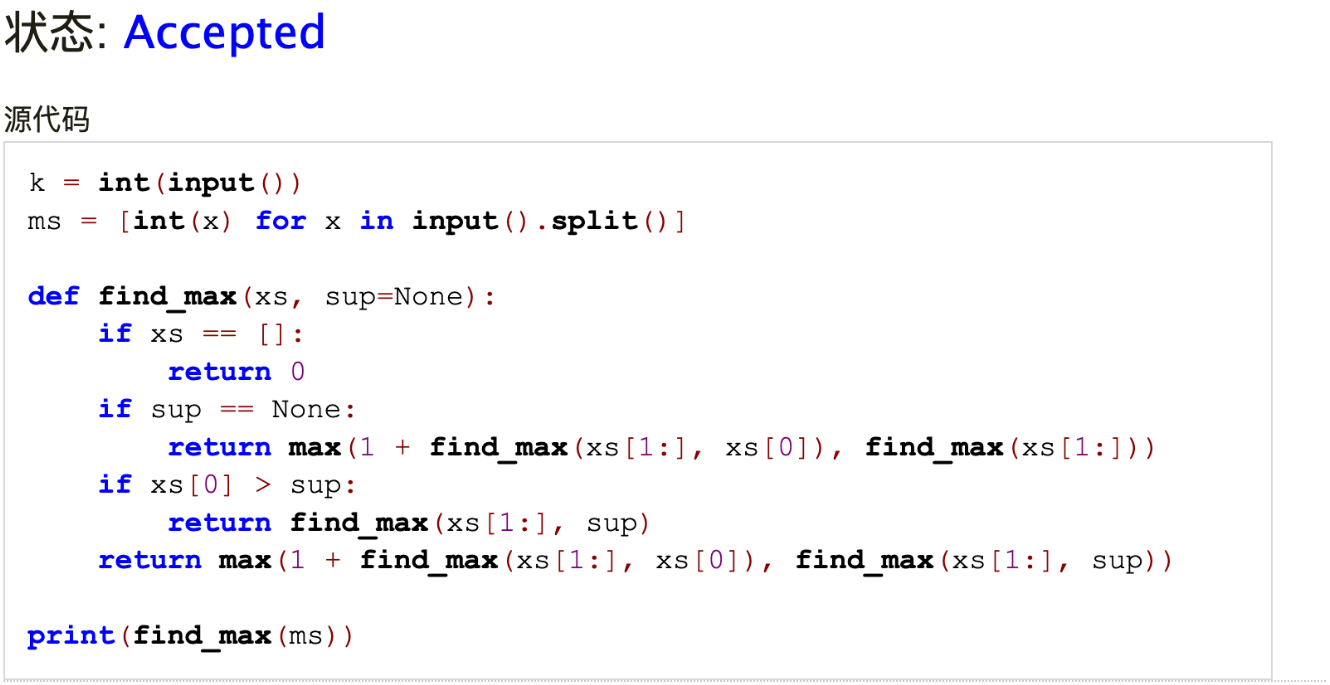
思路：

时隔一年，动态规划有些不熟悉了，在考场上使用了递归解法。

代码

```python
k = int(input())
ms = [int(x) for x in input().split()]

def find_max(xs, sup=None):
    if xs == []:
        return 0
    if sup == None:
        return max(1 + find_max(xs[1:], xs[0]), find_max(xs[1:]))
    if xs[0] > sup:
        return find_max(xs[1:], sup)
    return max(1 + find_max(xs[1:], xs[0]), find_max(xs[1:], sup))

print(find_max(ms))
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```python
k = int(input())
ms = [int(x) for x in input().split()]

def find_max(xs, sup=None):
    if xs == []:
        return 0
    if sup == None:
        return max(1 + find_max(xs[1:], xs[0]), find_max(xs[1:]))
    if xs[0] > sup:
        return find_max(xs[1:], sup)
    return max(1 + find_max(xs[1:], xs[0]), find_max(xs[1:], sup))

print(find_max(ms))
```

04147:汉诺塔问题(Tower of Hanoi)

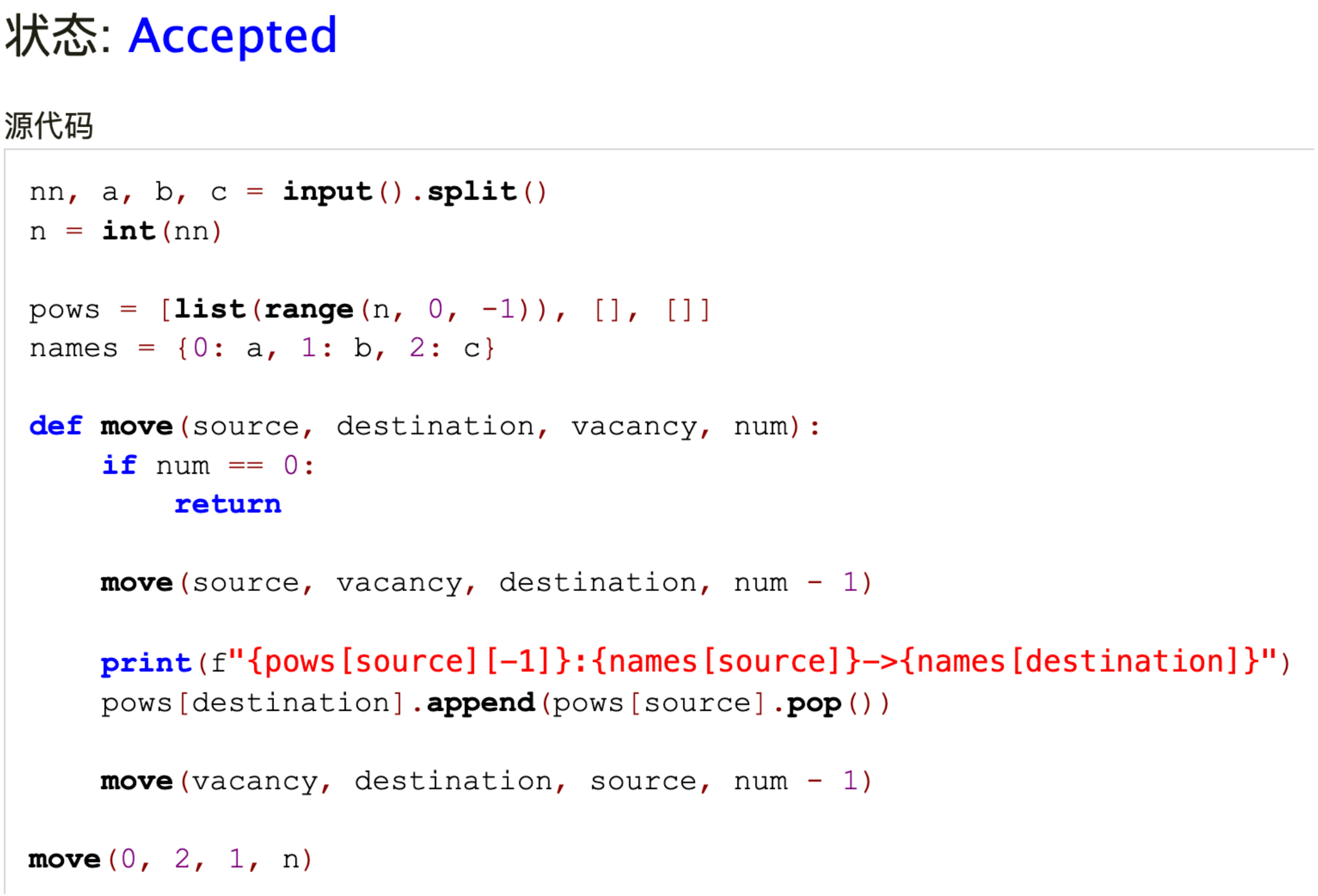
http://cs101.openjudge.cn/practice/04147

思路：根据题目提示，使用递归

代码

```python
nn, a, b, c = input().split()
n = int(nn)

pows = [list(range(n, 0, -1)), [], []]
names = {0: a, 1: b, 2: c}

def move(source, destination, vacancy, num):
    if num == 0:
        return

    move(source, vacancy, destination, num - 1)

    print(f"{pows[source][-1]}:{names[source]}->{names[destination]}")
    pows[destination].append(pows[source].pop())

    move(vacancy, destination, source, num - 1)

move(0, 2, 1, n)
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```python
nn, a, b, c = input().split()
n = int(nn)

pows = [list(range(n, 0, -1)), [], []]
names = {0: a, 1: b, 2: c}

def move(source, destination, vacancy, num):
    if num == 0:
        return

    move(source, vacancy, destination, num - 1)

    print(f"{pows[source][-1]}:{names[source]}->{names[destination]}")
    pows[destination].append(pows[source].pop())

    move(vacancy, destination, source, num - 1)

move(0, 2, 1, n)
```

03253: 约瑟夫问题No.2

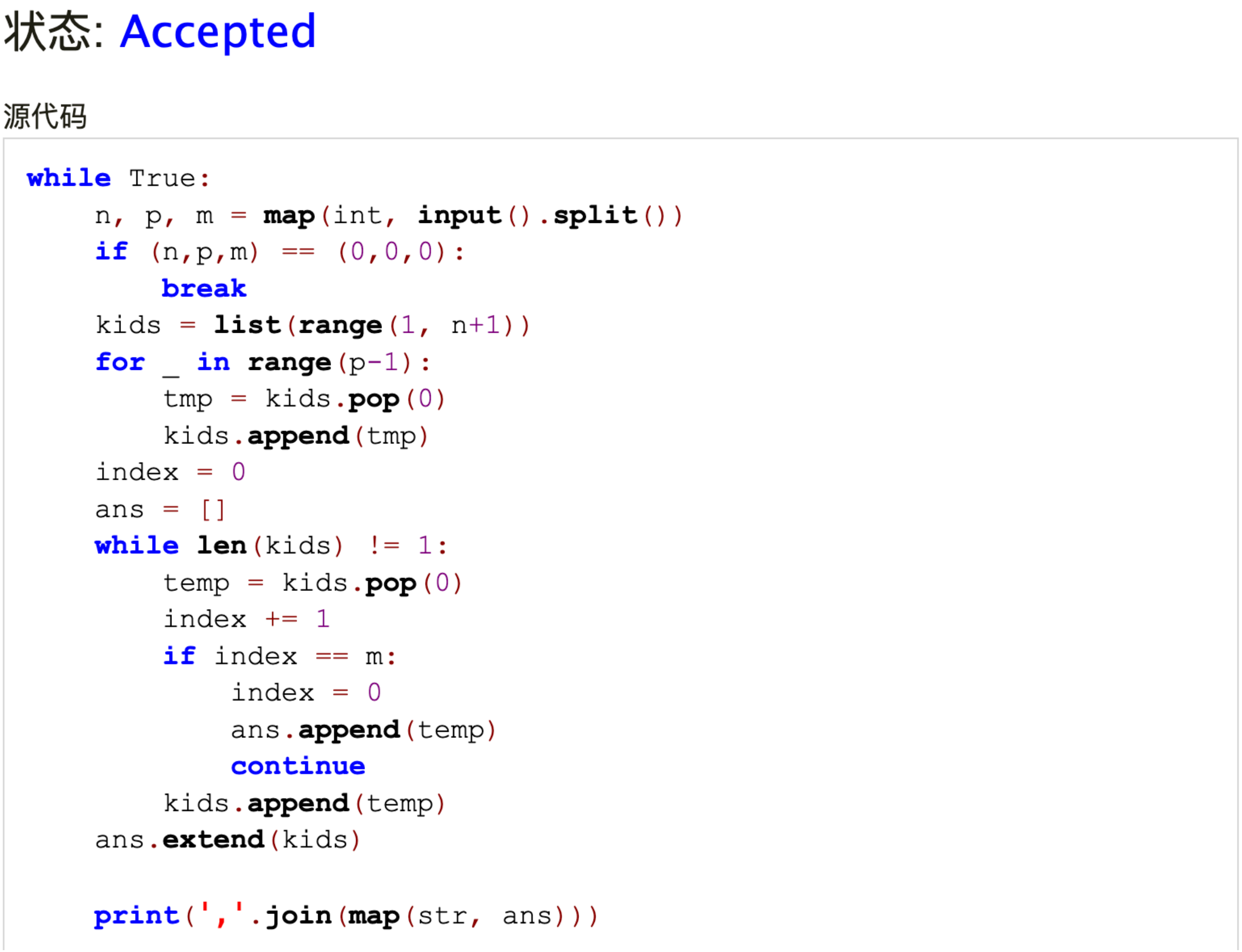
http://cs101.openjudge.cn/practice/03253

思路：直接模拟过程实现

代码

```
while True:
    n, p, m = map(int, input().split())
    if (n,p,m) == (0,0,0):
        break
    kids = list(range(1, n+1))
    for _ in range(p-1):
        tmp = kids.pop(0)
        kids.append(tmp)
    index = 0
    ans = []
    while len(kids) != 1:
        temp = kids.pop(0)
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        kids.append(temp)
    ans.extend(kids)

    print(','.join(map(str, ans)))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
while True:
    n, p, m = map(int, input().split())
    if (n,p,m) == (0,0,0):
        break
    kids = list(range(1, n+1))
    for _ in range(p-1):
        tmp = kids.pop(0)
        kids.append(tmp)
    index = 0
    ans = []
    while len(kids) != 1:
        temp = kids.pop(0)
        index += 1
        if index == m:
            index = 0
            ans.append(temp)
            continue
        kids.append(temp)
    ans.extend(kids)

    print(','.join(map(str, ans)))
```

21554:排队做实验 (greedy)v0.2

http://cs101.openjudge.cn/practice/21554

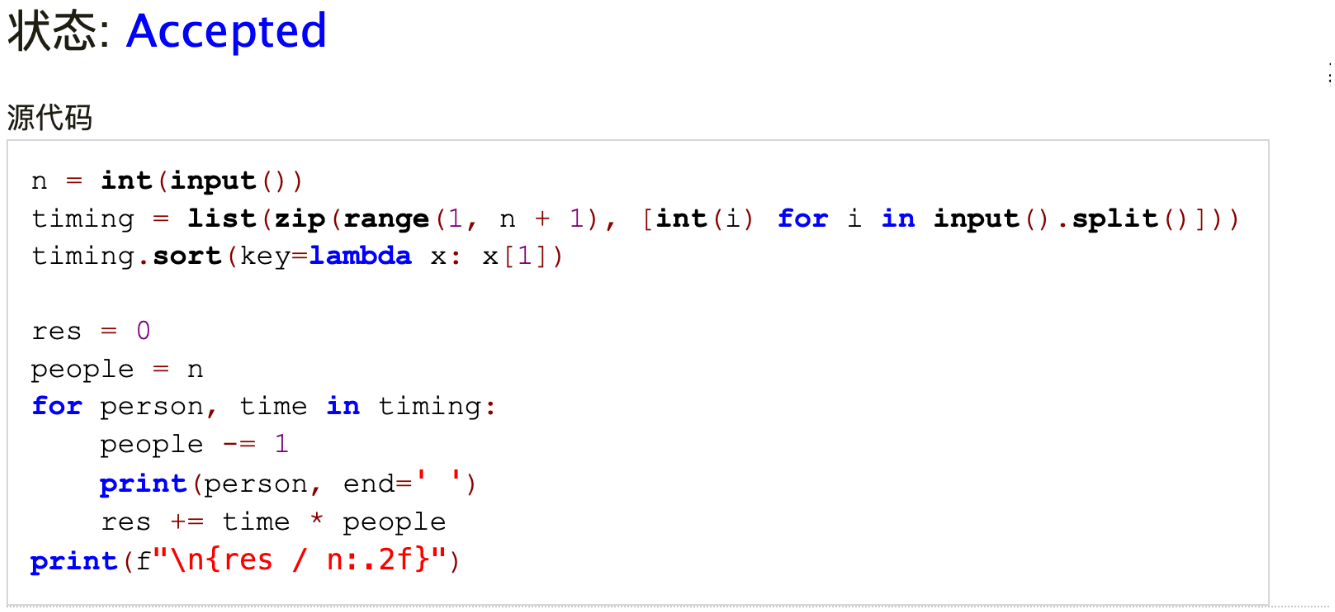
思路：

贪心，按时间排序后遍历

代码

```
n = int(input())
timing = list(zip(range(1, n + 1), [int(i) for i in input().split()]))
timing.sort(key=lambda x: x[1])

res = 0
people = n
for person, time in timing:
    people -= 1
    print(person, end=' ')
    res += time * people
print(f"\n{res / n:.2f}")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
n = int(input())
timing = list(zip(range(1, n + 1), [int(i) for i in input().split()]))
timing.sort(key=lambda x: x[1])

res = 0
people = n
for person, time in timing:
    people -= 1
    print(person, end=' ')
    res += time * people
print(f"\n{res / n:.2f}")
```

19963:买学区房

http://cs101.openjudge.cn/practice/19963

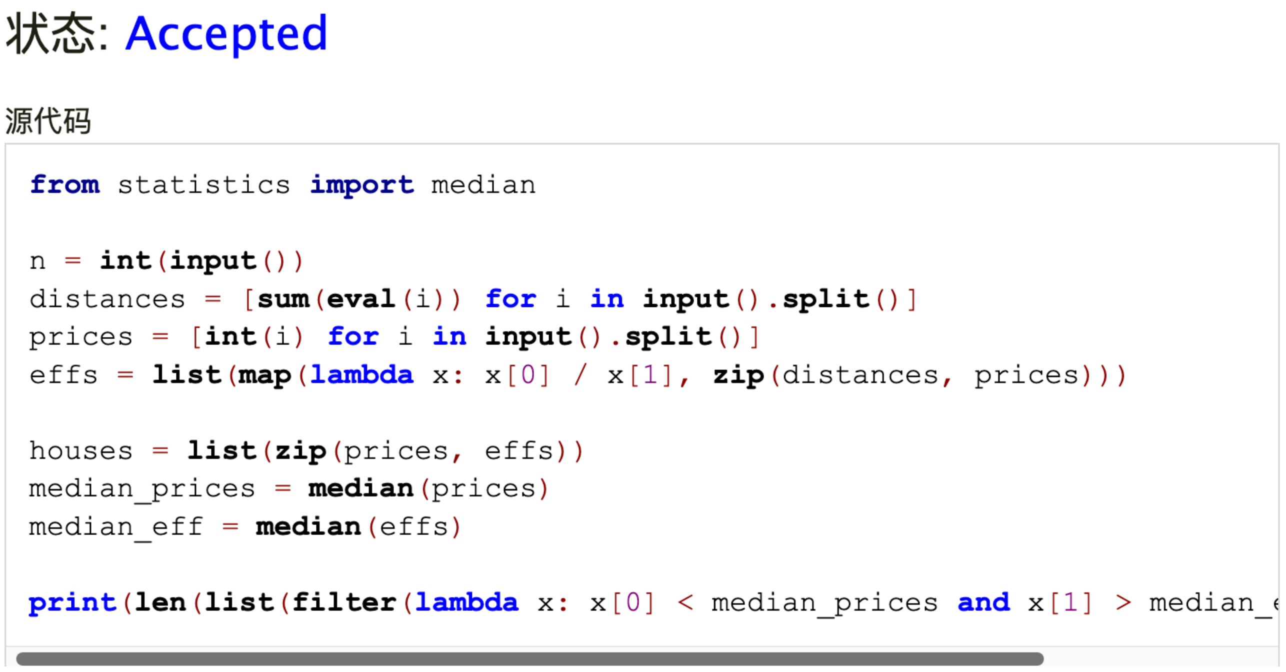
思路：

正常求解，模拟题目的要求

代码

```python
from statistics import median

n = int(input())
distances = [sum(eval(i)) for i in input().split()]
prices = [int(i) for i in input().split()]
effs = list(map(lambda x: x[0] / x[1], zip(distances, prices)))

houses = list(zip(prices, effs))
median_prices = median(prices)
median_eff = median(effs)

print(len(list(filter(lambda x: x[0] < median_prices and x[1] > median_eff, houses))))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```python
from statistics import median

n = int(input())
distances = [sum(eval(i)) for i in input().split()]
prices = [int(i) for i in input().split()]
effs = list(map(lambda x: x[0] / x[1], zip(distances, prices)))

houses = list(zip(prices, effs))
median_prices = median(prices)
median_eff = median(effs)

print(len(list(filter(lambda x: x[0] < median_prices and x[1] > median_e
```

27300: 模型整理

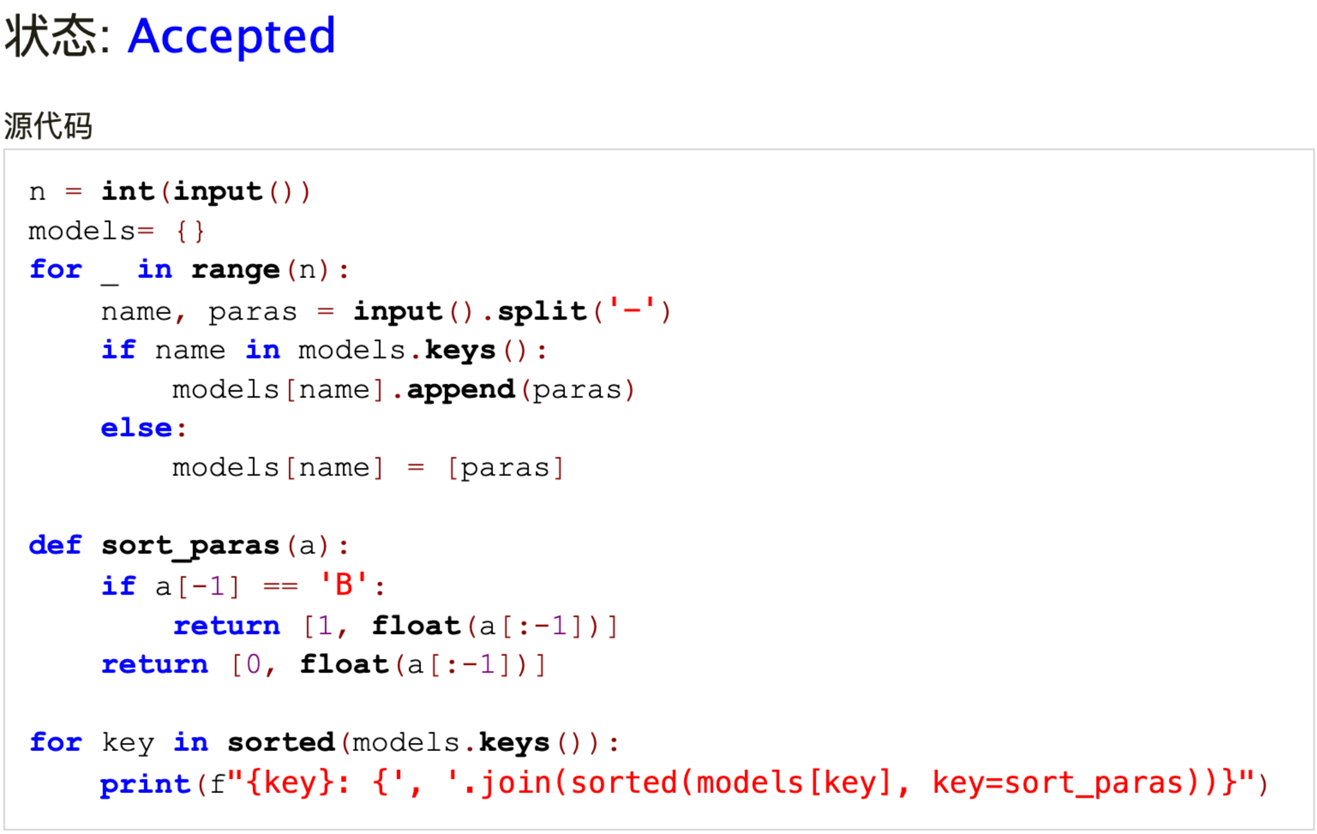
http://cs101.openjudge.cn/practice/27300

思路：正常的字符串处理

代码

```python
n = int(input())
models= {}
for _ in range(n):
    name, paras = input().split('-')
    if name in models.keys():
        models[name].append(paras)
    else:
        models[name] = [paras]

def sort_paras(a):
    if a[-1] == 'B':
        return [1, float(a[:-1])]
    return [0, float(a[:-1])]

for key in sorted(models.keys()):
    print(f"{key}: {', '.join(sorted(models[key], key=sort_paras))}")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```python
n = int(input())
models= {}
for _ in range(n):
    name, paras = input().split('-')
    if name in models.keys():
        models[name].append(paras)
    else:
        models[name] = [paras]

def sort_paras(a):
    if a[-1] == 'B':
        return [1, float(a[:-1])]
    return [0, float(a[:-1])]

for key in sorted(models.keys()):
    print(f"{key}: {', '.join(sorted(models[key], key=sort_paras))}")
```

2. 学习总结和收获

==如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。==

整体难度不太大，不过在考场上做约瑟夫问题时没有使用直接模拟的方法，结果WA看不到样例，debug未果，AC5。