Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by 潘子轩、信科

说明：

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），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"区有上传的md或者doc附件。

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

02808: 校门外的树

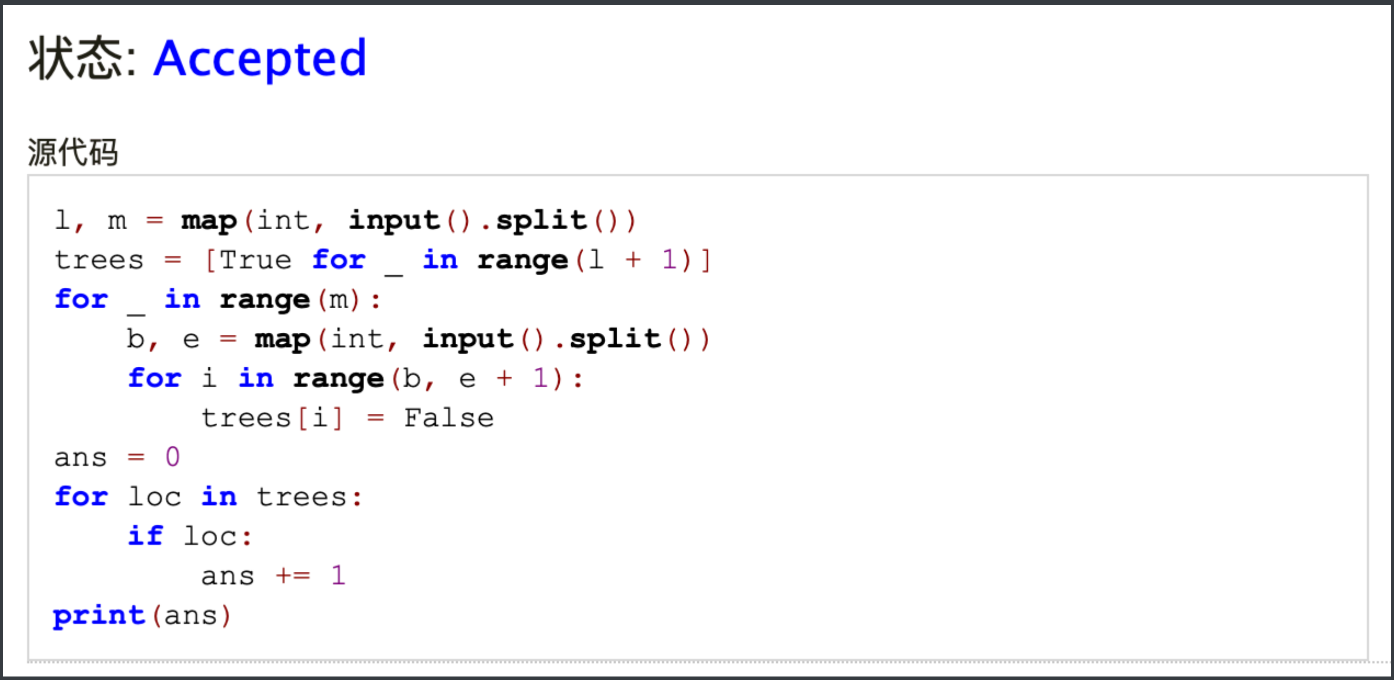
http://cs101.openjudge.cn/practice/02808/

思路：

代码

```python
l, m = map(int, input().split())
trees = [True for _ in range(l + 1)]
for _ in range(m):
    b, e = map(int, input().split())
    for i in range(b, e + 1):
        trees[i] = False
ans = 0
for loc in trees:
    if loc:
        ans += 1
print(ans)
```

代码运行截图 （至少包含有"Accepted"）


状态: Accepted

源代码

```python
l, m = map(int, input().split())
trees = [True for _ in range(l + 1)]
for _ in range(m):
    b, e = map(int, input().split())
    for i in range(b, e + 1):
        trees[i] = False
ans = 0
for loc in trees:
    if loc:
        ans += 1
print(ans)
```

20449: 是否被5整除

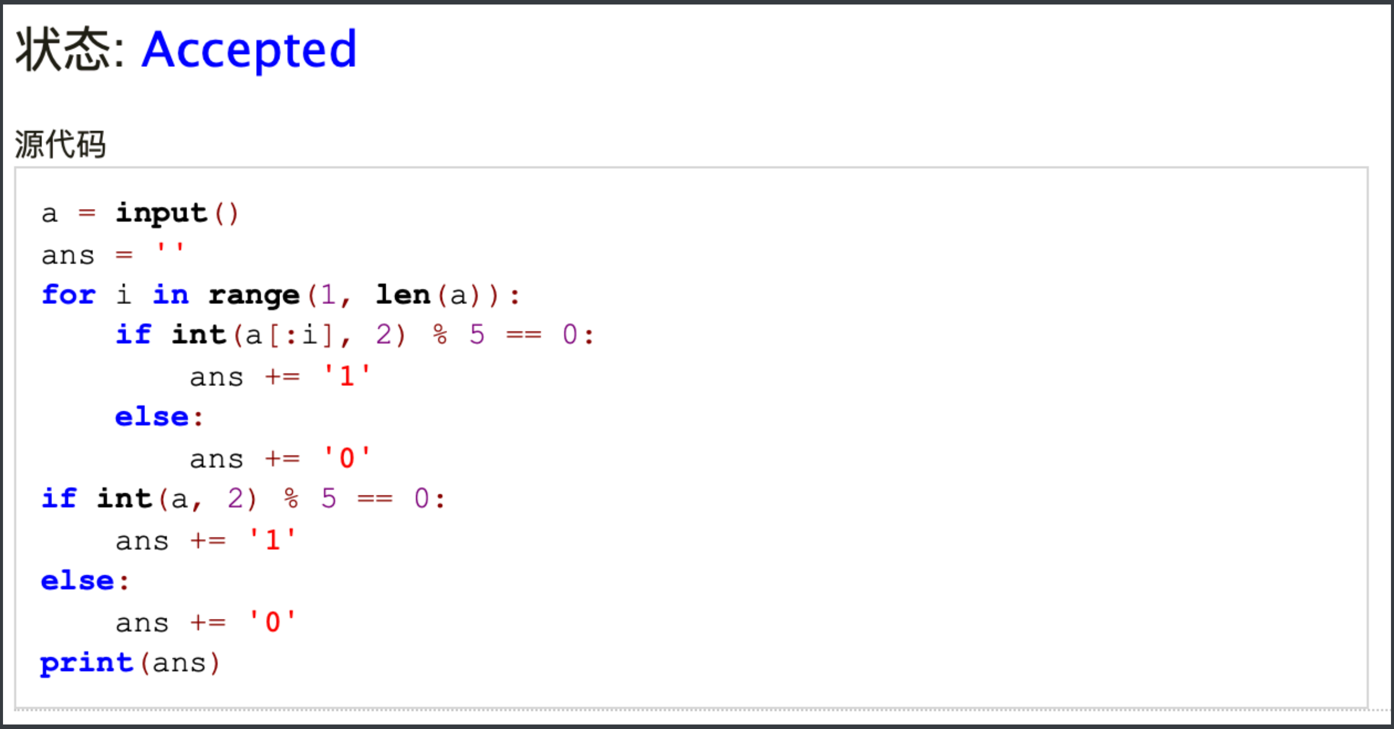
http://cs101.openjudge.cn/practice/20449/

思路：

代码

```python
a = input()
ans = ''
for i in range(1, len(a)):
    if int(a[:i], 2) % 5 == 0:
        ans += '1'
    else:
        ans += '0'
if int(a, 2) % 5 == 0:
    ans += '1'
else:
    ans += '0'
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
a = input()
ans = ''
for i in range(1, len(a)):
    if int(a[:i], 2) % 5 == 0:
        ans += '1'
    else:
        ans += '0'
if int(a, 2) % 5 == 0:
    ans += '1'
else:
    ans += '0'
print(ans)
```

01258: Agri-Net

http://cs101.openjudge.cn/practice/01258/

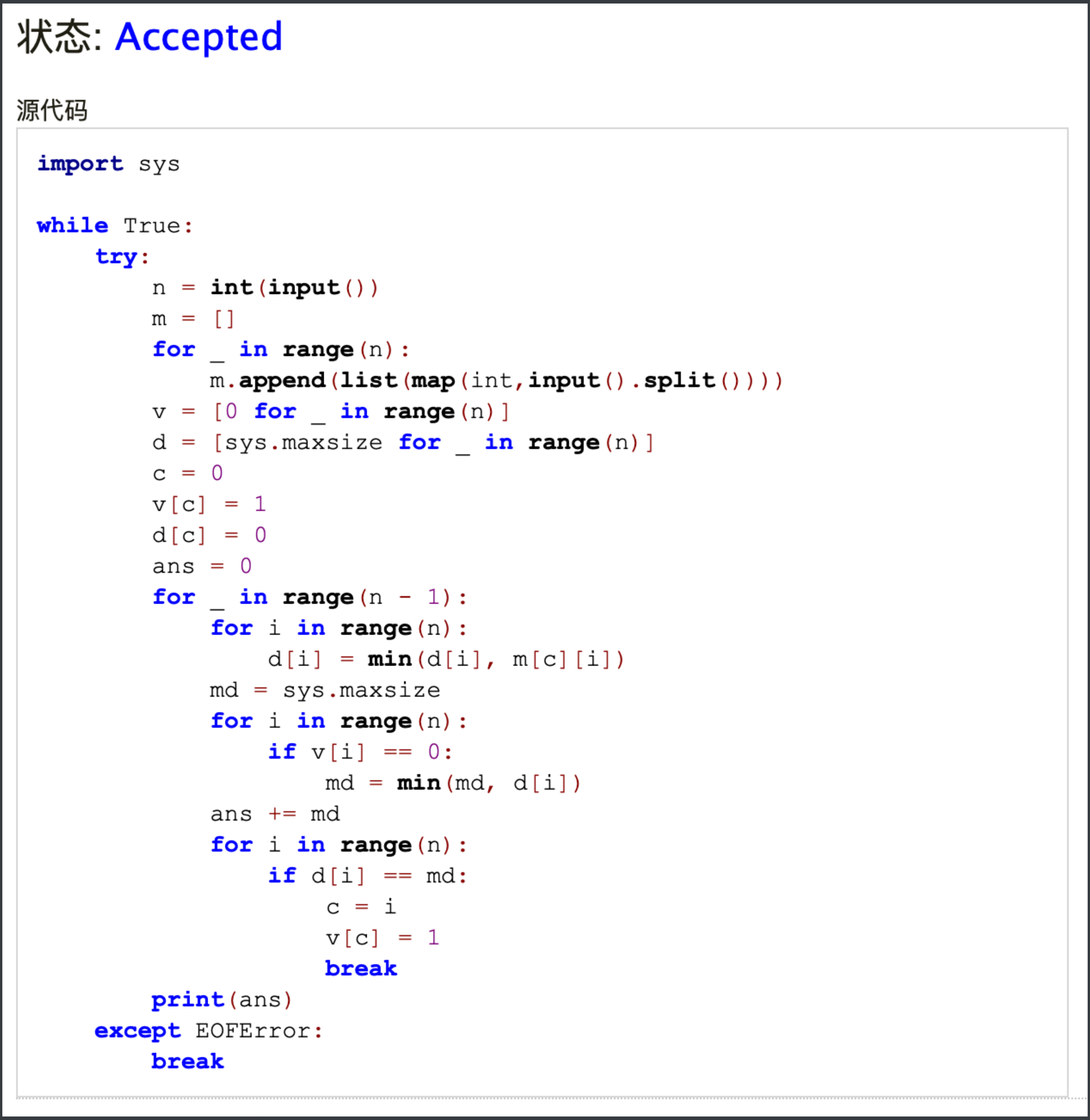
思路：

代码

```
import sys

while True:
    try:
        n = int(input())
        m = []
        for _ in range(n):
            m.append(list(map(int,input().split())))
        v = [0 for _ in range(n)]
        d = [sys.maxsize for _ in range(n)]
        c = 0
        v[c] = 1
        d[c] = 0
        ans = 0
        for _ in range(n - 1):
            for i in range(n):
                d[i] = min(d[i], m[c][i])
            md = sys.maxsize
            for i in range(n):
                if v[i] == 0:
                    md = min(md, d[i])
            ans += md
            for i in range(n):
                if d[i] == md:
                    c = i
                    v[c] = 1
                    break
        print(ans)
    except EOFError:
        break
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
import sys

while True:
    try:
        n = int(input())
        m = []
        for _ in range(n):
            m.append(list(map(int,input().split())))
        v = [0 for _ in range(n)]
        d = [sys.maxsize for _ in range(n)]
        c = 0
        v[c] = 1
        d[c] = 0
        ans = 0
        for _ in range(n - 1):
            for i in range(n):
                d[i] = min(d[i], m[c][i])
            md = sys.maxsize
            for i in range(n):
                if v[i] == 0:
                    md = min(md, d[i])
            ans += md
            for i in range(n):
                if d[i] == md:
                    c = i
                    v[c] = 1
                    break
        print(ans)
    except EOFError:
        break
```

27635: 判断无向图是否连通有无回路(同23163)

http://cs101.openjudge.cn/practice/27635/

思路：

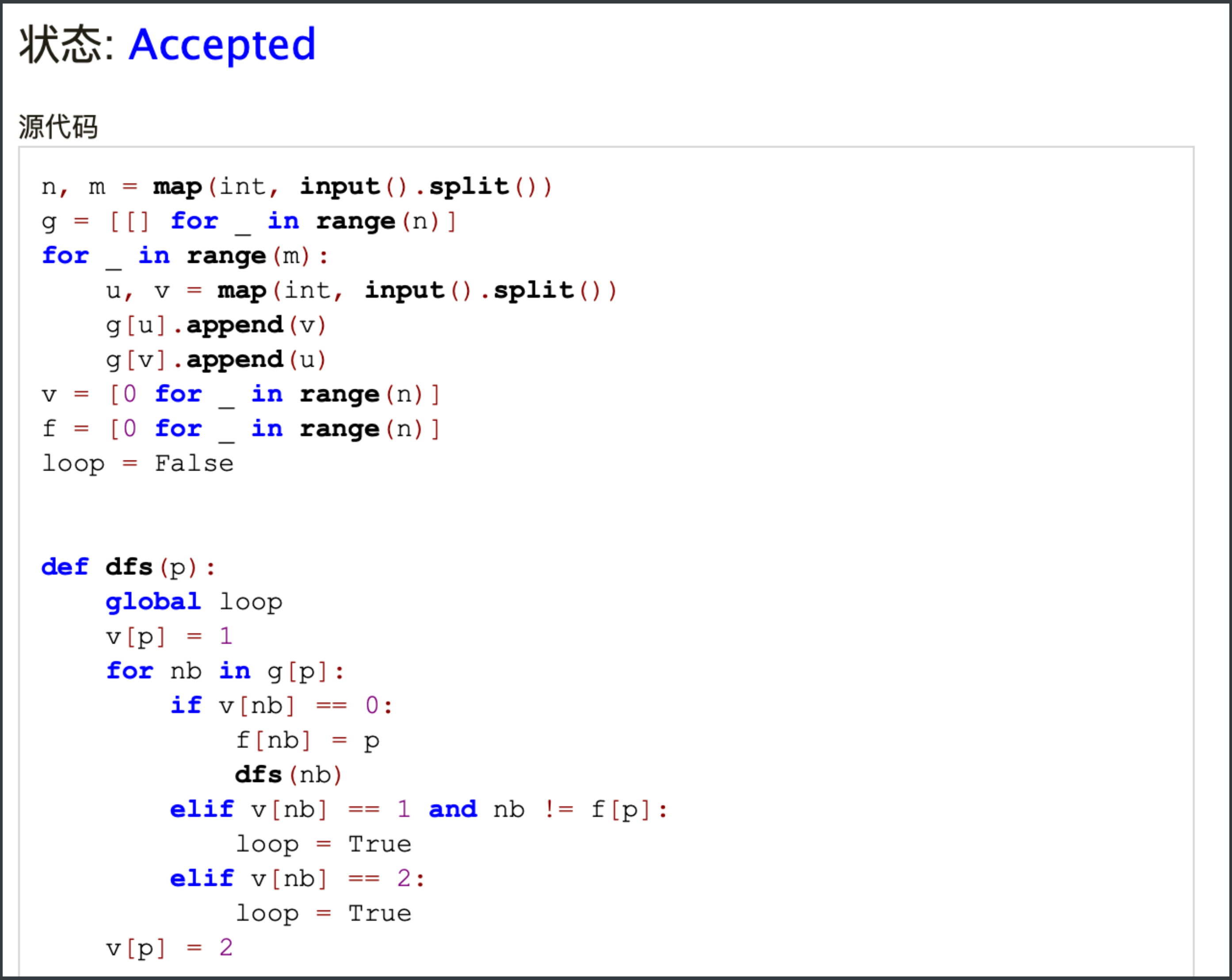
代码

```
n, m = map(int, input().split())
g = [[] for _ in range(n)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v)
    g[v].append(u)
v = [0 for _ in range(n)]
f = [0 for _ in range(n)]
loop = False


def dfs(p):
    global loop
    v[p] = 1
    for nb in g[p]:
        if v[nb] == 0:
            f[nb] = p
            dfs(nb)
        elif v[nb] == 1 and nb != f[p]:
            loop = True
        elif v[nb] == 2:
            loop = True
    v[p] = 2
```

```python
connected = True
for i in range(n):
    if v[i] == 0:
        dfs(i)
    if i == 0:
        for ii in v:
            if ii != 2:
                connected = False
if connected:
    print('connected:yes')
else:
    print('connected:no')
if loop:
    print('loop:yes')
else:
    print('loop:no')
```

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, m = map(int, input().split())
g = [[] for _ in range(n)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v)
    g[v].append(u)
v = [0 for _ in range(n)]
f = [0 for _ in range(n)]
loop = False


def dfs(p):
    global loop
    v[p] = 1
    for nb in g[p]:
        if v[nb] == 0:
            f[nb] = p
            dfs(nb)
        elif v[nb] == 1 and nb != f[p]:
            loop = True
        elif v[nb] == 2:
            loop = True
    v[p] = 2
```

27947: 动态中位数

http://cs101.openjudge.cn/practice/27947/

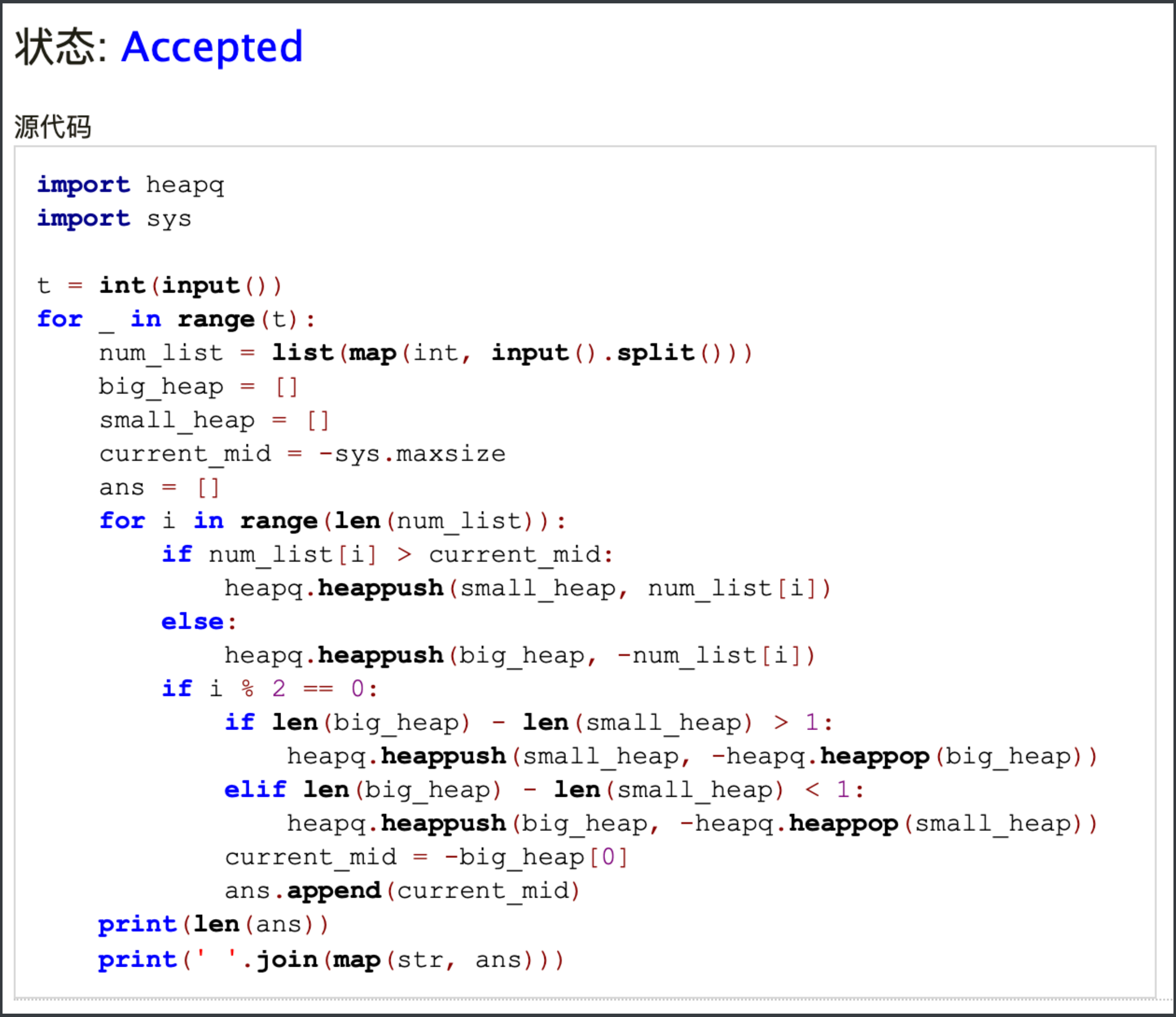
思路：用一个最大堆一个最小堆来“顶”出来中位数。

代码

```python
import heapq
import sys

t = int(input())
for _ in range(t):
    num_list = list(map(int, input().split()))
    big_heap = []
    small_heap = []
    current_mid = -sys.maxsize
    ans = []
    for i in range(len(num_list)):
        if num_list[i] > current_mid:
            heapq.heappush(small_heap, num_list[i])
        else:
            heapq.heappush(big_heap, -num_list[i])
        if i % 2 == 0:
            if len(big_heap) - len(small_heap) > 1:
                heapq.heappush(small_heap, -heapq.heappop(big_heap))
            elif len(big_heap) - len(small_heap) < 1:
                heapq.heappush(big_heap, -heapq.heappop(small_heap))
            current_mid = -big_heap[0]
            ans.append(current_mid)
    print(len(ans))
    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

```python
import heapq
import sys

t = int(input())
for _ in range(t):
    num_list = list(map(int, input().split()))
    big_heap = []
    small_heap = []
    current_mid = -sys.maxsize
    ans = []
    for i in range(len(num_list)):
        if num_list[i] > current_mid:
            heapq.heappush(small_heap, num_list[i])
        else:
            heapq.heappush(big_heap, -num_list[i])
        if i % 2 == 0:
            if len(big_heap) - len(small_heap) > 1:
                heapq.heappush(small_heap, -heapq.heappop(big_heap))
            elif len(big_heap) - len(small_heap) < 1:
                heapq.heappush(big_heap, -heapq.heappop(small_heap))
            current_mid = -big_heap[0]
            ans.append(current_mid)
    print(len(ans))
    print(' '.join(map(str, ans)))
```

28190: 奶牛排队

http://cs101.openjudge.cn/practice/28190/

思路：用两个单调栈，二分查找。

代码

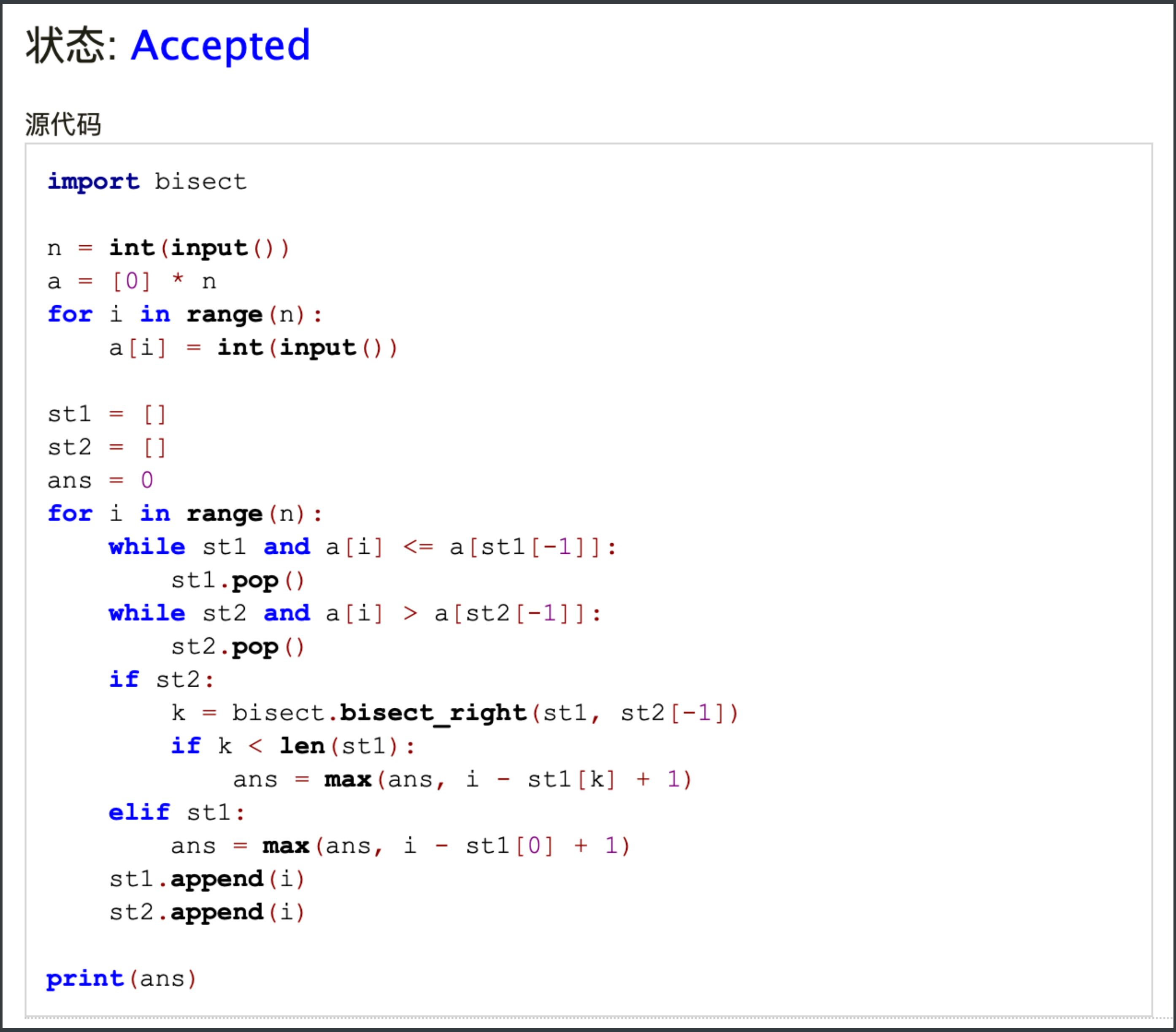
```python
import bisect

n = int(input())
a = [0] * n
for i in range(n):
    a[i] = int(input())

st1 = []
st2 = []
ans = 0
for i in range(n):
    while st1 and a[i] <= a[st1[-1]]:
        st1.pop()
    while st2 and a[i] > a[st2[-1]]:
        st2.pop()
    if st2:
        k = bisect.bisect_right(st1, st2[-1])
        if k < len(st1):
            ans = max(ans, i - st1[k] + 1)
    elif st1:
        ans = max(ans, i - st1[0] + 1)
    st1.append(i)
    st2.append(i)

print(ans)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）


状态: Accepted

源代码

```python
import bisect

n = int(input())
a = [0] * n
for i in range(n):
    a[i] = int(input())

st1 = []
st2 = []
ans = 0
for i in range(n):
    while st1 and a[i] <= a[st1[-1]]:
        st1.pop()
    while st2 and a[i] > a[st2[-1]]:
        st2.pop()
    if st2:
        k = bisect.bisect_right(st1, st2[-1])
        if k < len(st1):
            ans = max(ans, i - st1[k] + 1)
    elif st1:
        ans = max(ans, i - st1[0] + 1)
    st1.append(i)
    st2.append(i)

print(ans)
```

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。

后两道题都要用到一些比较巧妙的方法，还是得多练多见。