

# Assignment #D: May月考

---

Updated 1654 GMT+8 May 8, 2024

2024 spring, Compiled by 天幕 化学与分子工程学院

## 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

操作系统: Windows 11 23H2

Python编程环境: Visual Studio Code 1.86.2230.

## 1. 题目

---

### 02808: 校门外的树

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

思路: 使用set去重

代码

```
1 s = set()
2 x, y = map(int, input().split())
3 for i in range(y):
4     a, b = map(int, input().split())
5     for j in range(a, b + 1):
6         s.add(j)
7 print(x + 1 - len(s))
```

代码运行截图

状态: Accepted

源代码

```
s = set()
x, y = map(int, input().split())
for i in range(y):
    a, b = map(int, input().split())
    for j in range(a, b + 1):
        s.add(j)
print(x + 1 - len(s))
```

## 20449: 是否被5整除

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

思路: 基础题

代码

```
1 x = input()
2 c = 0
3 ans = ''
4 for i in x:
5     c *= 2
6     c += int(i)
7     if c % 5 == 0:
8         ans += '1'
9     else:
10        ans += '0'
11 print(ans)
```

代码运行截图

状态: Accepted

源代码

```
x = input()
c = 0
ans = ''
for i in x:
    c *= 2
    c += int(i)
    if c % 5 == 0:
        ans += '1'
    else:
        ans += '0'
print(ans)
```

©2002-2022 POJ 京ICP备20010980号-1

## 01258: Agri-Net

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

思路：看了很久才发现会有多组数据.....希望期末考试不要在审题上太恐怖

代码

```
1  from heapq import *
2
3  while True:
4      try:
5          n = int(input())
6          temp = []
7          l = [list(map(int, input().split())) for _ in range(n)]
8          visited = [0]
9          unvisited = list(range(1, n))
10         ans = 0
11         ways = []
12         for onode in unvisited:
13             heappush(ways, (l[0][onode], 0, onode))
14         while unvisited:
15             way = heappop(ways)
16             weight, this, other = way
17             if other in visited:
18                 continue
19             ans += weight
20             visited.append(other)
21             unvisited.remove(other)
22             for onode in unvisited:
23                 heappush(ways, (l[other][onode], other, onode))
```

```
24     print(ans)
25 except EOFError:
26     break
```

代码运行截图

状态: Accepted

源代码

```
from heapq import *

while True:
    try:
        n = int(input())
        temp = []
        l = [list(map(int, input().split())) for _ in range(n)]
        visited = [0]
        unvisited = list(range(1, n))
        ans = 0
        ways = []
        for onode in unvisited:
            heappush(ways, (l[0][onode], 0, onode))
        while unvisited:
            way = heappop(ways)
            weight, this, other = way
            if other in visited:
                continue
            ans += weight
            visited.append(other)
            unvisited.remove(other)
            for onode in unvisited:
                heappush(ways, (l[other][onode], other, onode))
        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: 图论基础题, 遍历并上色

代码

```
1 n, m = map(int, input().split())
2 l = [[False] * n for _ in range(n)]
3 true = [[True] * n for _ in range(n)]
4 for _ in range(m):
```

```
5     a, b = map(int, input().split())
6     l[a][b] = l[b][a] = True
7
8     connected = 'yes'
9     loop = 'no'
10    black = []
11    grey = []
12    q = [0]
13    while q:
14        c = q.pop()
15        if c in black:
16            continue
17        grey.append(c)
18        for i in range(n):
19            if l[c][i]:
20                if i in black:
21                    continue
22                if i in grey and i:
23                    loop = 'yes'
24                    continue
25                q.append(i)
26                grey.append(i)
27        black.append(c)
28        f = c
29    if len(black) != n:
30        connected = 'no'
31    print('connected:{}\nloop:{}'.format(connected, loop))
```

代码运行截图

## 状态: Accepted

源代码

```
n, m = map(int, input().split())
l = [[False] * n for _ in range(n)]
true = [[True] * n for _ in range(n)]
for _ in range(m):
    a, b = map(int, input().split())
    l[a][b] = l[b][a] = True

connected = 'yes'
loop = 'no'
black = []
grey = []
q = [0]
while q:
    c = q.pop()
    if c in black:
        continue
    grey.append(c)
    for i in range(n):
        if l[c][i]:
            if i in black:
                continue
            if i in grey and i:
                loop = 'yes'
                continue
            q.append(i)
            grey.append(i)
    black.append(c)
    f = c
if len(black) != n:
    connected = 'no'
print('connected: {}\nloop: {}'.format(connected, loop))
```

## 27947: 动态中位数

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

思路: 维护两个堆分别存储中位数前后

代码

```
1 from heapq import *
2
3 for _ in range(int(input())):
4     lb = []
```

```

5     ls = []
6     count = 0
7     shouldprint = True
8     ans = []
9     for x in map(int, input().split()):
10         if not lb or -lb[0] >= x:
11             heappush(lb, -x)
12         else:
13             heappush(ls, x)
14             if len(ls) > len(lb):
15                 heappush(lb, -heappop(ls))
16             if len(lb) > len(ls) + 1:
17                 heappush(ls, -heappop(lb))
18             if shouldprint:
19                 ans.append(str(-lb[0]))
20             shouldprint = not shouldprint
21     print(str(len(ans)) + '\n' + ' '.join(ans))

```

代码运行截图

状态: **Accepted**

源代码

```

from heapq import *

for _ in range(int(input())):
    lb = []
    ls = []
    count = 0
    shouldprint = True
    ans = []
    for x in map(int, input().split()):
        if not lb or -lb[0] >= x:
            heappush(lb, -x)
        else:
            heappush(ls, x)
            if len(ls) > len(lb):
                heappush(lb, -heappop(ls))
            if len(lb) > len(ls) + 1:
                heappush(ls, -heappop(lb))
            if shouldprint:
                ans.append(str(-lb[0]))
            shouldprint = not shouldprint
    print(str(len(ans)) + '\n' + ' '.join(ans))

```

## 28190: 奶牛排队

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

思路：好难，看了各种题解才看明白。维护索引为*i*的数右边第一个小于等于其的数的索引，再维护一个左边第一个大于其的。

代码

```
1  n = int(input())
2  l = [float('inf')] + [int(input()) for _ in range(n)] + [0]
3  rmin = dict()
4  lmax = dict()
5  stackmin = [0]
6  stackmax = [n + 1]
7  for i in range(1, n + 2):
8      while len(stackmin) > 1 and l[stackmin[-1]] >= l[i]:
9          rmin[stackmin.pop()] = i
10     stackmin.append(i)
11  for i in range(n, -1, -1):
12     while len(stackmax) > 1 and l[stackmax[-1]] <= l[i]:
13         lmax[stackmax.pop()] = i
14     stackmax.append(i)
15  ans = 0
16  for i in range(1, n + 1):
17     for j in range(i + 1, rmin[i]):
18         if i > lmax[j]:
19             ans = max(ans, j - i + 1)
20  print(ans)
```

代码运行截图



状态: Accepted

源代码

```
n = int(input())
l = [float('inf')] + [int(input()) for _ in range(n)] + [0]
rmin = dict()
lmax = dict()
stackmin = [0]
stackmax = [n + 1]
for i in range(1, n + 2):
    while len(stackmin) > 1 and l[stackmin[-1]] >= l[i]:
        rmin[stackmin.pop()] = i
    stackmin.append(i)
for i in range(n, -1, -1):
    while len(stackmax) > 1 and l[stackmax[-1]] <= l[i]:
        lmax[stackmax.pop()] = i
    stackmax.append(i)
ans = 0
for i in range(1, n + 1):
    for j in range(i + 1, rmin[i]):
        if i > lmax[j]:
            ans = max(ans, j - i + 1)
print(ans)
```

## 2. 学习总结和收获

最后一题好难，看了好久的各类题解才看懂，需要预习了233