

# Assignment #D: May月考

---

Updated 1654 GMT+8 May 8, 2024

2024 spring, Compiled by ==田济维 物理学院

## 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

(python pycharm)

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

```

1  #
2  L,M = map(int,input().split())
3  tree = [0]*(L+2)
4  for i in range(M):
5      start,end = map(int,input().split())
6      tree[start]+=1
7      tree[end+1]-=1
8  for i in range(L+1):
9      tree[i+1]+=tree[i]
10 print(tree.count(0)-1)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

L,M = map(int,input().split())
tree = [0]*(L+2)
for i in range(M):
    start,end = map(int,input().split())
    tree[start]+=1
    tree[end+1]-=1
for i in range(L+1):
    tree[i+1]+=tree[i]
print(tree.count(0)-1)

```

©2002-2022 POJ 京ICP备20010980号-1

## 20449: 是否被5整除

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

思路:

代码

```

1  # s = input()
2  cnt = 0
3
4  for x in s:
5      cnt=cnt*2+int(x)
6      if cnt%5==0:
7          print("1",end = "")
8      else:
9          print("0",end="")
10

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
s = input()
cnt = 0

for x in s:
    cnt=cnt*2+int(x)
    if cnt%5==0:
        print("1",end="")
    else:
        print("0",end="")
```

©2002-2022 POJ 京ICP备20010980号-1

## 01258: Agri-Net

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

思路:

代码

```
1 #
2 from heapq import *
3 while True:
4     try:
5         n = int(input())
6     except EOFError:
7         break
8     else:
9         graph = [[0] * n for i in range(n)]
10        ru = []
11        for i in range(n):
12            ru.extend(list(map(int, input().split())))
13        for i in range(n):
14            graph[i][:] = ru[n * i:n * i + n]
15        mst = []
16        used = set([0])
17        edges = [(graph[0][i], 0, i) for i in range(1, n)]
18        heapify(edges)
19        cnt = 0
20        while edges:
21            cost, frm, to = heappop(edges)
22            if to not in used:
23                used.add(to)
24                cnt += cost
```

```

25         for i in range(n):
26             if i != to and i not in used:
27                 heappush(edges, (graph[to][i], to, i))
28     print(cnt)
29

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```

from heapq import *
while True:
    try:
        n = int(input())
    except EOFError:
        break
    else:
        graph = [[0] * n for i in range(n)]
        ru = []
        for i in range(n):
            ru.extend(list(map(int, input().split())))
        for i in range(n):
            graph[i][:] = ru[n * i:n * i + n]
        mst = []
        used = set([0])
        edges = [(graph[0][i], 0, i) for i in range(1, n)]
        heapify(edges)
        cnt = 0
        while edges:
            cost, frm, to = heappop(edges)
            if to not in used:

```

基本信息

#: 44  
 题目: 01  
 提交人: 23  
 内存: 45  
 时间: 42  
 语言: Py3  
 提交时间: 20

## 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

```

1  #
2  n,m = map(int,input().split())
3  graph = [[] for i in range(n)]
4  for i in range(m):
5      s,e = map(int,input().split())
6      graph[s].append(e)
7      graph[e].append(s)
8  flag = 0
9
10 tag = 0
11 def dfs(i,parent=-1):
12     global flag,visited,tag

```

```

13     visited.add(i)
14     for j in graph[i]:
15         if j in visited and j!=parent:
16             flag = 1
17         elif j not in visited:
18             dfs(j,i)
19 for i in range(n):
20     visited = set()
21     dfs(i,-1)
22     if len(visited)==n:
23         tag = 1
24         if flag == 1:
25             break
26
27 if tag:
28     print("connected:yes")
29 else:
30     print("connected:no")
31
32 if flag:
33     print("loop:yes")
34 else:
35     print("loop:no")
36
37

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```

n,m = map(int,input().split())
graph = [[] for i in range(n)]
for i in range(m):
    s,e = map(int,input().split())
    graph[s].append(e)
    graph[e].append(s)
flag = 0

tag = 0
def dfs(i,parent=-1):
    global flag,visited,tag
    visited.add(i)
    for j in graph[i]:
        if j in visited and j!=parent:
            flag = 1
        elif j not in visited:
            dfs(j,i)
for i in range(n):
    visited = set()
    dfs(i,-1)

```

## 27947: 动态中位数

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

思路:

代码

```
1  #
2  from heapq import *
3  nu = int(input())
4  for i in range(nu):
5      s = list(map(int, input().split()))
6      n = len(s)
7      print((n+1)//2)
8      minh = []
9      maxh = []
10     middle = s[0]
11     for i in range(n):
12         if s[i]>middle:
13             heappush(maxh, s[i])
14         elif s[i]<=middle:
15             heappush(minh, -s[i])
16         if i%2==0:
17             if len(minh)>len(maxh)+1:
18                 heappush(maxh, -heappop(minh))
19             elif len(minh)<len(maxh):
20                 heappush(minh, -heappop(maxh))
21             middle = -heappop(minh)
22             print(middle, end=" ")
23             heappush(minh, -middle)
24     print("")
25
26
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
from heapq import *
nu = int(input())
for i in range(nu):
    s = list(map(int, input().split()))
    n = len(s)
    print((n+1)//2)
    minh = []
    maxh = []
    middle = s[0]
    for i in range(n):
        if s[i]>middle:
            heappush(maxh, s[i])
        elif s[i]<=middle:
            heappush(minh, -s[i])
        if i%2==0:
            if len(minh)>len(maxh)+1:
                heappush(maxh, -heappop(minh))
            elif len(minh)<len(maxh):
                heappush(minh, -heappop(maxh))
            middle = -heappop(minh)
            print(middle, end=" ")
            heappush(minh, -middle)
    print("")
```

## 28190: 奶牛排队

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

思路: 不会做, 学习了单调栈

代码

```
1  #
2  N = int(input())
3  heights = [int(input()) for _ in range(N)]
4
5  left_bound = [-1] * N
6  right_bound = [N] * N
7
8  stack = [] # 单调栈, 存储索引
9
10 # 求左侧第一个≥h[i]的奶牛位置
11 for i in range(N):
12     while stack and heights[stack[-1]] < heights[i]:
13         stack.pop()
14
15     if stack:
16         left_bound[i] = stack[-1]
```

```

17     stack.append(i)
18
19
20 stack = [] # 清空栈以供寻找右边界使用
21
22 # 求右侧第一个 $\leq h[i]$ 的奶牛位
23 for i in range(N-1, -1, -1):
24     while stack and heights[stack[-1]] > heights[i]:
25         stack.pop()
26
27     if stack:
28         right_bound[i] = stack[-1]
29
30     stack.append(i)
31
32 ans = 0
33
34 # for i in range(N-1, -1, -1): # 从大到小枚举是个技巧
35 #     for j in range(left_bound[i] + 1, i):
36 #         if right_bound[j] > i:
37 #             ans = max(ans, i - j + 1)
38 #             break
39 #
40 #     if i <= ans:
41 #         break
42
43 for i in range(N): # 枚举右端点 B寻找 A, 更新 ans
44     for j in range(left_bound[i] + 1, i):
45         if right_bound[j] > i:
46             ans = max(ans, i - j + 1)
47             break
48 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

```
N = int(input())
heights = [int(input()) for _ in range(N)]

left_bound = [-1] * N
right_bound = [N] * N

stack = [] # 单调栈, 存储索引

# 求左侧第一个≥h[i]的奶牛位置
for i in range(N):
    while stack and heights[stack[-1]] < heights[i]:
        stack.pop()

    if stack:
        left_bound[i] = stack[-1]

    stack.append(i)
```

## 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。

学习了单调栈的性质