

# Assignment #D: May月考

---

Updated 1654 GMT+8 May 8, 2024

2024 spring, Compiled by 余汶青 生命科学学院

## 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: 版本 Windows 11 家庭中文版

版本 22H2

安装日期 2023/7/18

操作系统版本 22621.2283

序列号 5CD323PJKL

体验 Windows Feature Experience Pack 1000.22662.1000.0

Python编程环境: \* Spyder version: 5.4.3 (conda)

- Python version: 3.11.4 64-bit
- Qt version: 5.15.2
- PyQt5 version: 5.15.7
- Operating System: Windows 10

## 1. 题目

---

### 02808: 校门外的树

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

思路:

代码

```

L,M=map(int,input().split())
tree=[1]*(L+1)
summ=0

for i in range(M):
    a,b=map(int,input().split())
    for j in range(b-a+1):
        tree[j+a]=0

for i in range(L+1):
    summ+=tree[i]
print(summ)

```

代码运行截图

#41237794提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

L,M=map(int,input().split())
tree=[1]*(L+1)
summ=0

```

基本信息

#: 41237794  
 题目: 02808  
 提交人: 23n2300012265  
 内存: 3616kB  
 时间: 63ms

## 20449: 是否被5整除

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

思路:

代码

```

s=input()
total=0
for i in s:
    total=total*2+int(i)
    if total%5==0:
        print(1,end='')
    else:
        print(0,end='')
print()

```

代码运行截图

状态: Accepted

源代码

```
s=input()
total=0
for i in s:
    total=total*2+int(i)
    if total%5==0:
        print(1,end='')
    else:
```

基本信息

#: 44931605  
题目: 20449  
提交人: 23n2300012265  
内存: 3660kB  
时间: 23ms  
语言: Python3  
提交时间: 2024-05-11 18:55:48

## 01258: Agri-Net

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

思路:

代码

```
class DisJointSet():
    def __init__(self,n):
        self.father={}
        self.rank={}
        for i in range(n):
            self.father[i]=i
            self.rank[i]=0
    def find(self,x):
        if self.father[x]!=x:
            self.father[x]=self.find(self.father[x])
        return self.father[x]
    def union(self,x,y):
        rootx=self.find(x)
        rooty=self.find(y)
        if rootx!=rooty:
            if self.rank[rootx]>self.rank[rooty]:
                self.father[rooty]=rootx
            elif self.rank[rootx]<self.rank[rooty]:
                self.father[rootx]=rooty
            else:
                self.father[rootx]=rooty
                self.rank[rooty]+=1

while True:
    try:
        n=int(input())
        edge=[]
        for i in range(n):
            s=[int(k) for k in input().split()]
            for j in range(i,n):
                edge.append((s[j],i,j))

        edge.sort()
        disjoint_set=DisJointSet(n)
        ans=0
```

```

for weight,x,y in edge:
    if disjoint_set.find(x)!=disjoint_set.find(y):
        #print(weight,x,y)
        disjoint_set.union(x,y)
        ans+=weight
print(ans)
except:
    break

```

代码运行截图

#44954720提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class DisJointSet():
    def __init__(self,n):
        self.father={}
        self.rank={}
        for i in range(n):
            self.father[i]=i
            self.rank[i]=0

```

基本信息

#: 44954720  
 题目: 01258  
 提交人: 23n2300012265  
 内存: 4080kB  
 时间: 44ms  
 语言: Python3  
 提交时间: 2024-05-13 20:41:21

## 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

```

class DisJointSet():
    def __init__(self,n):
        self.father={}
        self.rank={}
        for i in range(n):
            self.father[i]=i
            self.rank[i]=0
    def find(self,x):
        if self.father[x]!=x:
            self.father[x]=self.find(self.father[x])
        return self.father[x]
    def union(self,x,y):
        rootx=self.find(x)
        rooty=self.find(y)
        if rootx!=rooty:
            if self.rank[rootx]>self.rank[rooty]:
                self.father[rooty]=rootx
            elif self.rank[rootx]<self.rank[rooty]:
                self.father[rootx]=rooty
            else:
                self.father[rootx]=rooty

```

```

        self.rank[rooty]+=1

n,m=map(int,input().split())
disjoint_set=DisJointSet(n)
v1=0
v2=0
for i in range(m):
    x,y=map(int,input().split())
    if disjoint_set.find(x)==disjoint_set.find(y):
        v2=1
    else:
        disjoint_set.union(x, y)
root=disjoint_set.find(0)
for i in range(1,n):
    if disjoint_set.find(i)!=root:
        v1=1
if v1:
    print("connected:no")
else:
    print("connected:yes")
if v2:
    print("loop:yes")
else:
    print("loop:no")

```

代码运行截图

#44954816提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class DisJointSet():
    def __init__(self,n):
        self.father={}
        self.rank={}
        for i in range(n):
            self.father[i]=i
            self.rank[i]=0
    def find(self,x):

```

基本信息

#: 44954816  
 题目: 27635  
 提交人: 23n2300012265  
 内存: 3704kB  
 时间: 31ms  
 语言: Python3  
 提交时间: 2024-05-13 20:48:32

## 27947: 动态中位数

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

思路:

代码

```

import heapq

T=int(input())
for _ in range(T):

```

```

a=[int(i) for i in input().split()]
heapleft=[100000]
heapright=[100000]
mid=a[0]
if len(a)%2==0:
    a.pop()
print((len(a)+1)//2)
for i in range(1,len(a),2):
    print(mid,end=' ')
    s=[mid,-heapq.heappop(heapleft),heapq.heappop(heapright),a[i],a[i+1]]
    s.sort()
    #print(i,s)
    mid=s[2]
    heapq.heappush(heapleft,-s[0])
    heapq.heappush(heapleft,-s[1])
    heapq.heappush(heapright,s[3])
    heapq.heappush(heapright,s[4])
print(mid)

```

代码运行截图

#44955094提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

import heapq

T=int(input())
for _ in range(T):
    a=[int(i) for i in input().split()]
    heapleft=[100000]
    heapright=[100000]
    mid=a[0]

```

基本信息

#: 44955094  
 题目: 27947  
 提交人: 23n2300012265  
 内存: 10920kB  
 时间: 352ms  
 语言: Python3  
 提交时间: 2024-05-13 21:09:12

## 28190: 奶牛排队

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

思路:

代码

```
# 熊江凯、元培学院
from bisect import bisect_right as bl
lis,q1,q2,ans=[int(input())for _ in range(int(input()))],[-1],[-1],0
for i in range(len(lis)):
    while len(q1)>1 and lis[q1[-1]]>=lis[i]:q1.pop()
    while len(q2)>1 and lis[q2[-1]]<lis[i]:q2.pop()
    id=bl(q1,q2[-1])
    if id<len(q1):ans=max(ans,i-q1[id]+1)
    q1.append(i)
    q2.append(i)
print(ans)
```

代码运行截图

#44957059提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
# 熊江凯、元培学院
from bisect import bisect_right as bl
lis,q1,q2,ans=[int(input())for _ in range(int(input()))],[-1],[-1],0
for i in range(len(lis)):
    while len(q1)>1 and lis[q1[-1]]>=lis[i]:q1.pop()
    while len(q2)>1 and lis[q2[-1]]<lis[i]:q2.pop()
    id=bl(q1,q2[-1])
    if id<len(q1):ans=max(ans,i-q1[id]+1)
    q1.append(i)
    q2.append(i)
print(ans)
```

基本信息

#: 44957059  
题目: 28190  
提交人: 23n2300012265  
内存: 39468kB  
时间: 2278ms  
语言: Python3  
提交时间: 2024-05-14 00:18:42

## 2. 学习总结和收获

题目是后来自己做的

一二题很简单

三四题都需要是图的模板题，但是做的时候发现忘记了最小生成树该怎么求，于是又去翻之前的代码，说明基本算法掌握还是不够熟练，需要多练习

第五题用堆，上学期数算练了很多堆的题，所以还算简单

最后一道题看了半天没有思路，然后看题解，新学了单调栈这种数据结构，不过即便知道单调栈是什么不看题解我自己应该也想不到解法，题解实在太巧妙了