Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by ==夏天、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

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows 10 家庭版

Python编程环境：Spyder (python 3.11)

1. 题目

02808: 校门外的树

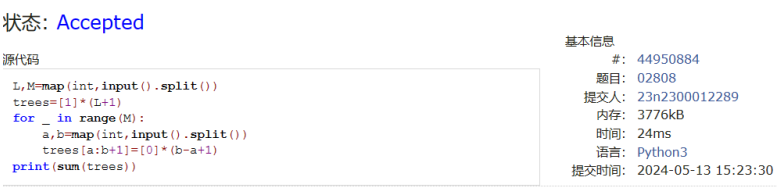
http://cs101.openjudge.cn/practice/02808/

思路:上学期的摸底考试题(好像是),很简单就不赘述了

代码

```
L,M=map(int,input().split())
trees=[1]*(L+1)
for _ in range(M):
    a,b=map(int,input().split())
    trees[a:b+1]=[0]*(b-a+1)
print(sum(trees))
```

代码运行截图 ==（至少包含有"Accepted"）==



20449: 是否被5整除

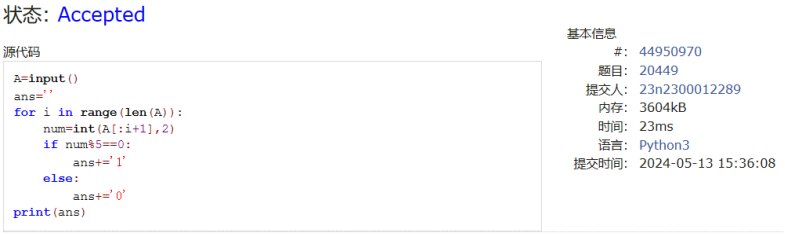
http://cs101.openjudge.cn/practice/20449/

思路：用int(' ',base)的第二个参数转成对应进制的数之后就很简单了

代码

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

代码运行截图 ==（至少包含有"Accepted"）==



01258: Agri-Net

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

思路：兔子与星空同类题，用Prim算法

代码

```
while True:
    try:
        N=int(input())
        graph={}
        for i in range(N):
            distances=list(map(int,input().split()))
            graph[i+1]={}
            for j,distance in enumerate(distances):
                if j!=i:
                    graph[i+1][j+1]=distance
        total=0
        visited=set()
        visited.add(1)
        while len(visited)<len(graph):
            min_distance=float('inf')
            min_farm=None
            for farm in visited:
                for next_farm in graph[farm]:
                    if next_farm not in visited:
                        if graph[farm][next_farm]<min_distance:
                            min_distance=graph[farm][next_farm]
                            min_farm=next_farm
            if min_farm:
                total+=min_distance
                visited.add(min_farm)
        print(total)
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

27635: 判断无向图是否连通有无回路(同23163)

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

思路：模板题，是否连通就看能否访问到全部节点；有无回路就看访问邻居时是否访问到除相连节点以外的已访问节点。

代码

```
n,m=map(int,input().split())
graph={i:[] for i in range(n)}
for j in range(m):
    a,b=map(int,input().split())
    graph[a].append(b)
    graph[b].append(a)
def is_connected(graph,n):
    visited=set()
    stack=[0]
    while stack:
        node=stack.pop()
        if node not in visited:
            visited.add(node)
            stack.extend(graph[node])
    return len(visited)==n
def has_loop(graph):
    visited=set()
    stack=[(0,-1)]
    while stack:
        node,parent=stack.pop()
```

```
            return True
        visited.add(node)
        for neighbor in graph[node]:
            if neighbor!=parent:
                stack.append((neighbor,node))
    return False
if is_connected(graph,n):
    print('connected:yes')
else:
    print('connected:no')
if has_loop(graph):
    print('loop:yes')
else:
    print('loop:no')
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

27947: 动态中位数

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

思路：用两个堆，且保持较大堆的元素个数

等于较小堆的元素个数或+1，每逢奇数个数

就输出较大堆中最小的数

代码

```
import heapq
T=int(input())
for _ in range(T):
    ans=[]
    min_heap=[]
    max_heap=[]
    nums=list(map(int,input().split()))
    for i in range(len(nums)):
        if not max_heap or nums[i]>max_heap[0]:
            heapq.heappush(max_heap,nums[i])
        else:
            heapq.heappush(min_heap,-nums[i])
        if len(max_heap)>len(min_heap)+1:
            heapq.heappush(min_heap,-heapq.heappop(max_heap))
        elif len(min_heap)>len(max_heap):
            heapq.heappush(max_heap,-heapq.heappop(min_heap))
        if i%2==0:
            ans.append(max_heap[0])
    print(len(ans))
    print(*ans)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

28190: 奶牛排队

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

思路：利用单调栈，left_bound用于记录以当前点为最右端，满足条件的最左端的索引减1；right_bound用于记录以当前节点为最左端，满足条件的最右端的索引加1，最终答案就是两段拼起来之后的最长长度

代码

```
N=int(input())
cows=[int(input())for i in range(N)]
left_bound=[-1]*N
right_bound=[N]*N
stack=[]
for j in range(N):
    while stack and cows[j]>cows[stack[-1]]:
        stack.pop()
    if stack:
        left_bound[j]=stack[-1]
    stack.append(j)
stack=[]
for k in range(N-1,-1,-1):
    while stack and cows[stack[-1]]>cows[k]:
        stack.pop()
    if stack:
        right_bound[k]=stack[-1]
    stack.append(k)
ans=0
for i in range(N):
    for j in range(left_bound[i]+1,i):
        if right_bound[j]>i:
            ans=max(ans,i-j+1)
            break
print(ans)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

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。==

上周因为要复习转院考试所以没有参加月考，事情结束之后今天找了2h计时完成，最终AC4.前两题属于签到题；Agri-net那道题一开始没细想用bfs去写了（可能是因为前两周做了比较多的bfs，dfs的题？），WA后发现不是“兔子与樱花”，而是“兔子与星空”，加之输入有多组测试数据和题面说一行数据会拆成多行而实际上并没有的奇怪现象，在这道题上浪费了比较多的时间（这里也提出个人的小建议：考试时希望对输入数据的格式能说明得更清楚一些）；M2是道模板题，但是写完之后基本上就没时间看两道T了；动态中位数有想到用两个堆，但不知道具体该怎么实现，后来才知道需要有平衡这么一段操作；奶牛排队提示要用单调栈，上学期写接雨水时自己学了这个，不过现在又忘了。
转系考试这件大事总算结束了，接下来就有时间复习笔试、整理cheat sheet、刷每日选做了，开心o(*￣▽￣*)ブ