Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by 城环 吴至超

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：看清题目边界

代码

```
#
L,M=map(int,input().split())
trees=[True for _ in range(L+1)]
for i in range(M):
    start,end=map(int,input().split())
    for m in range(start,end+1):
        if trees[m]:
            L-=1
            trees[m]=False

print(L+1)
```

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:

代码

```
#
A=input()
def convert(n):

    lis=list(n)
    num=0
    cnt=0
    for i in lis[::-1]:
        if i=="1":
            num+=2**cnt
            cnt+=1
```

```
        else:
            cnt+=1
    return num
temp=""
ans=[]
for g in range(len(A)):
    temp+=f"{A[g]}"
    new=convert(temp)
    if new%5==0:
        ans.append("1")
    else:
        ans.append("0")
print("".join(ans))
```

代码运行截图 ==（至少包含有"Accepted"）==

01258: Agri-Net

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

思路：krustal算法，并查集部分很重要

还要注意多组测试数据。。。

代码

```
#
import heapq
class unionandfind:
```

```
    def __init__(self,n):
        self.fathers=[int(i) for i in range(n)]
        self.height=[0]*n
    def find(self,a):
        if self.fathers[a]!=a:
            self.fathers[a]=self.find(self.fathers[a])
        return self.fathers[a]
    def union(self,a,b):#及时更新最深节点，避免某个节点被拉出去
        if self.find(a)!=self.find(b):
            if self.height[self.find(a)]>self.height[self.find(b)]:
                self.fathers[self.find(b)]=self.find(a)
                self.height[self.find(a)]+=1
            else:
                self.fathers[self.find(a)]=self.find(b)
                self.height[self.find(b)]+=1
while True:
    try:
        n=int(input())
        uandf=unionandfind(n)
        matrix=[]
        visited=set()
        for i in range(n):
            sample=[int(x) for x in input().split()]
            for m in range(n):
                if sample[m]!=0 and ((i,m)not in visited and (m,i) not in
visited):
                    heapq.heappush(matrix,(sample[m],i,m))
                    visited.add((i,m))
        search=set()
        ans=0
        while matrix:
            tempo=heapq.heappop(matrix)
            if uandf.find(tempo[1])!=uandf.find(tempo[2]):
                uandf.union(tempo[1],tempo[2])
                search.add(tempo[1])
                search.add(tempo[2])
                ans+=tempo[0]


            if len(search)==n:
                print(ans)
                break
    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：并查集or dfs

代码

```
# n,m=map(int,input().split())
lis=[[] for i in range(n)]
flag=1
for i in range(m):
    a, b = map(int, input().split())
    lis[a].append(b)
    lis[b].append(a)
vis=set()
def dfs(x,pre):
    global cnt,flag
    vis.add(x)
    for i in lis[x]:
        if i not in vis:
            dfs(i,x)
        elif i in vis and i!=pre:
            flag=0
dfs(0,None)
if len(vis)==n:
    print("connected:yes")
else:
    print("connected:no")
if flag==0:
    print("loop:yes")
```

```
else:
    print("loop:no")


    并查集
class unionandfind:
    def __init__(self,n):
        self.fathers=[i for i in range(n)]
    def find(self,a):
        if self.fathers[a]!=a:
            self.fathers[a]=self.find(self.fathers[a])
        return self.fathers[a]
    def union(self,a,b):
        a_fa=self.find(a)
        b_fa=self.find(b)
        if a_fa!=b_fa:
            self.fathers[a_fa]=b_fa
            return False
        else:
            return True

n,m=map(int,input().split())
uf=unionandfind(n)

flag=1
for u in range(m):
    a,b=map(int,input().split())
    if  uf.union(a,b):
        flag=0

uf.fathers=[uf.find(i) for i in uf.fathers]
uf.fathers=set(uf.fathers)
if len(uf.fathers)>1:
    print("connected:no")
else:
    print("connected:yes")
if flag==1:
    print("loop:no")
else:
    print("loop:yes")
```

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：最大堆和最小堆

代码

```
#
import heapq
def find(lis):
    maxheap=[]
    minheap=[]

    ans=[]
    for i,cnt in enumerate(lis):
        if not maxheap or cnt<=-maxheap[0]:
            heapq.heappush(maxheap,-cnt)
        else:
            heapq.heappush(minheap,cnt)
        if len(maxheap)-len(minheap)>1:
            heapq.heappush(minheap,-heapq.heappop(maxheap))
        elif len(minheap)-len(maxheap)>0:
            heapq.heappush(maxheap,-heapq.heappop(minheap))
        if i%2==0:
            ans.append(-maxheap[0])


    return ans
```

```
T=int(input())
for _ in range(T):
    sample=[int(x) for x in input().split()]
    a=find(sample)
    print(len(a))
    print(*a)
```

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：

代码

```
#
```

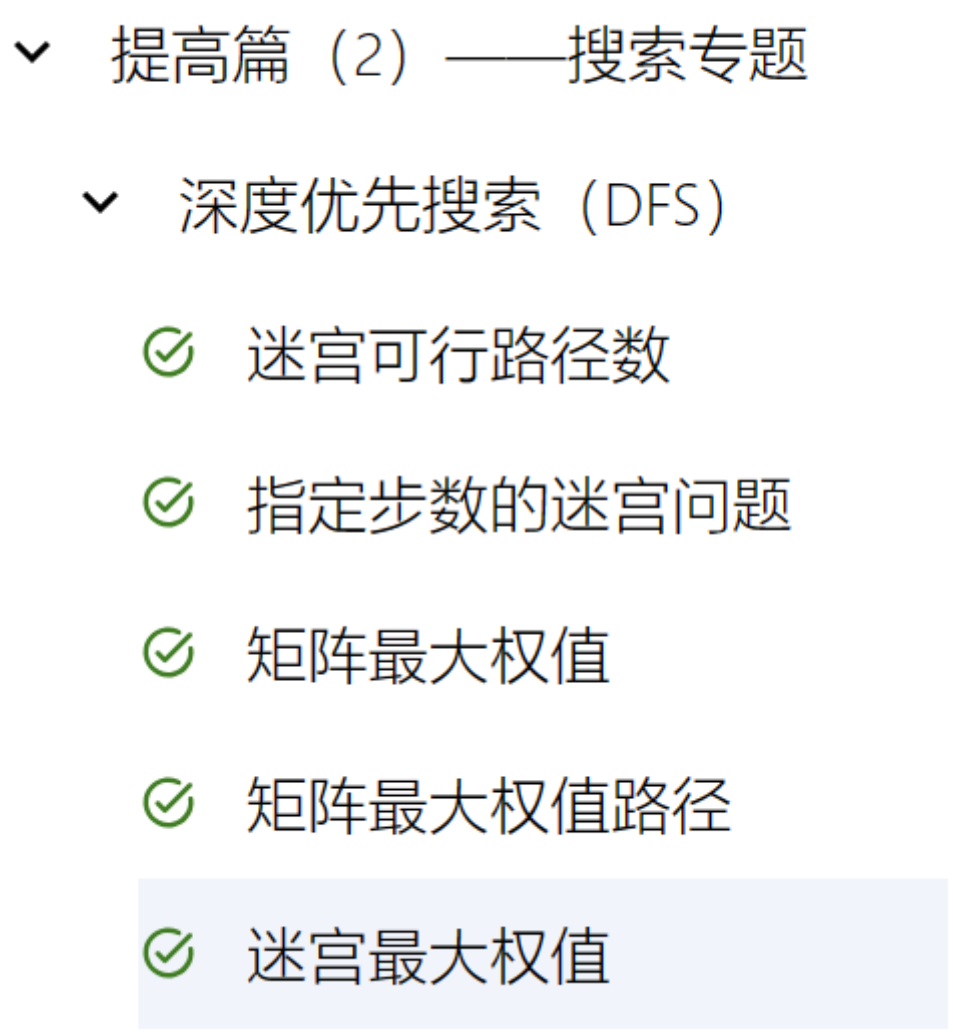
代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

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。==

感觉自己的bfs和dfs没有模板化，于是把晴问的题写了一遍

奶牛排队看反映好难的样子，先放放把模板题练好再说



- 提高篇（2）——搜索专题
  - 广度优先搜索（BFS）
    - 数字操作
    - 矩阵中的块
    - 迷宫问题
    - 迷宫最短路径
    - 跨步迷宫
    - 字符迷宫
    - 多终点迷宫问题
    - 迷宫问题-传送点
    - 中国象棋-马-无障碍
    - 中国象棋-马-有障碍