Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by 赵策 数学科学学院

编程环境

操作系统：Windows 11

Python编程环境：Visual Studio Code 1.86.2

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

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

代码运行截图 （至少包含有"Accepted"）

20449: 是否被5整除 Accepted

01258: Agri-Net

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

思路:

代码

```
#
def find_parent(i):
    if parent[i]==i:
        return i
    return find_parent(parent[i])

def union(x,y):
    rootX=find_parent(x)
    rootY=find_parent(y)
    parent[rootY]=rootX

def minmum_fibers(egdes):
    total_lenth=0
    for lenth,x,y in sorted(edges,key=lambda x:x[0]):
        rootX=find_parent(x)
        rootY=find_parent(y)
        if rootX!=rootY:
            total_lenth+=lenth
            union(rootX,rootY)
        if len(set(find_parent(i) for i in range(n)))==1:
```

```
            break
    return total_lenth

while True:
    try:
        n=int(input())
        edges=[]
        parent=list(range(n))
        for i in range(n):
            lenths=input().split()
            for j in range(n):
                lenth=int(lenths[j])
                edges.append((lenth,i,j))
        print(minmum_fibers(edges))
    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：

代码

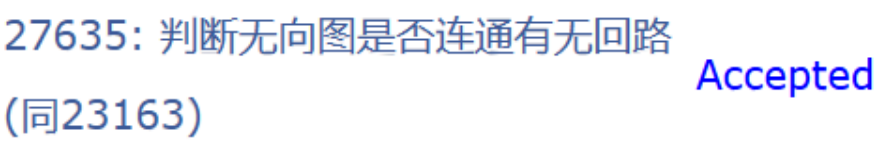
```
#
def is_connected():
    visited=[False]*n
    def dfs(node):
        visited[node]=True
        for neighbor in edges[node]:
            if not visited[neighbor]:
                dfs(neighbor)
    dfs(0)
    if all(visited):
        return 'connected:yes'
    else:
        return 'connected:no'

def has_loop():
    visited=[False]*n
    def dfs(node,parent,visited_set):
        visited[node]=True
        visited_set.add(node)
```

```
        for neighbor in edges[node]:
            if not visited[neighbor]:
                if dfs(neighbor,node,visited_set):
                    return True
            elif neighbor!=parent and neighbor in visited_set:
                return True
        visited_set.remove(node)
        return False
    for i in range(n):
        if not visited[i]:
            visited_set=set()
            if dfs(i,-1,visited_set):
                return 'loop:yes'
    return 'loop:no'

n,m=map(int,input().split())
edges=[[] for _ in range(n)]
for _ in range(m):
    u,v=map(int,input().split())
    edges[u].append(v)
    edges[v].append(u)
print(is_connected())
print(has_loop())
```

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：

代码

```
#
import heapq
def dynamic_median(nums):
    min_heap=[]
    max_heap=[]
    medians=[]
    for i,num in enumerate(nums):
        if len(max_heap)==0 or num<=-max_heap[0]:
            heapq.heappush(max_heap,-num)
```

```
        else:
            heapq.heappush(min_heap,num)
        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+1)%2==1:
            medians.append(-max_heap[0])
    return medians

for _ in range(int(input())):
    nums=list(map(int,input().split()))
    res=dynamic_median(nums)
    print(len(res))
    print(*res,sep=' ')
```

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

```
#
def sorted_cows(heights):
    left=[0]*n
    stack=[]
    for i in range(n):
        while stack and heights[stack[-1]]<heights[i]:
            stack.pop()
        left[i]=stack[-1]+1 if stack else 0
        stack.append(i)

    right=[n]*n
    stack=[]
    for i in range(n-1,-1,-1):
        while stack and heights[stack[-1]]>heights[i]:
            stack.pop()
        right[i]=stack[-1]+1 if stack else n
        stack.append(i)

    max_length=0
    for i in range(n-1,-1,-1):
```

```
        for j in range(left[i],i):
            if right[j]>i:
                max_length=max(max_length,i-j+1)
                break
        if i<=max_length:
            break
    return max_length

n=int(input())
heights=[int(input()) for _ in range(n)]
print(sorted_cows(heights))
```

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。

奶牛排队有点卡时间，最开始想用别的算法最后还是改用栈了