Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by 赵语涵 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：windows 11

Python编程环境：Spyder IDE 5.2.2

1. 题目

02808: 校门外的树

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

思路：是做过的题不过最开始完全没有想到说直接建一个大列表粗暴遍历。。。做的时候意外花了一些时间

代码

```
#赵语涵2300012254
l,m = map(int,input().split())
tree = [True for _ in range(l+1)]
for _ in range(m):
    s,e = map(int,input().split())
    for i in range(s,e+1):
        tree[i] = False
count = 0
for i in range(l+1):
    if tree[i]:
        count += 1
print(count)
```

代码运行截图

#45005305提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
l,m = map(int,input().split())
tree = [True for _ in range(l+1)]
for _ in range(m):
    s,e = map(int,input().split())
    for i in range(s,e+1):
```

基本信息

#: 45005305
题目: 02808
提交人: 23n2300012254
内存: 3912kB
时间: 43ms
语言: Python3
提交时间: 2024-05-18 19:49:35

20449: 是否被5整除

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

思路：题目很简单但是忘了进制字符等之间的互相转换了，还查了cheating paper才想起来

代码

```
#赵语涵2300012254
a = input()
ans = ''
for i in range(1,len(a)+1):
    x = int(str('0b'+a[:i]),2)
    if x%5 == 0:
        ans += '1'
    else:
        ans += '0'
print(ans)
```

代码运行截图

#44999687提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
a = input()
ans = ''
for i in range(1,len(a)+1):
    x = int(str('0b'+a[:i]),2)
    if x%5 == 0:
```

基本信息

#: 44999687
题目: 20449
提交人: 23n2300012254
内存: 3596kB
时间: 21ms
语言: Python3
提交时间: 2024-05-18 14:06:42

01258: Agri-Net

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

思路：依旧用了之前作业写过的Kruskal方法，第二次写感觉熟悉很多也写得很快（虽然考试应该可以直接抄模板吧。。。），不过最后WA苦恼地debug半天发现竟然是多组输入需要用try--except进行。。。还是要认真读题啊。。。

代码

```
#赵语涵2300012254
import heapq
def find(x):
    if parent[x] != x:
        return find(parent[x])
    return parent[x]
ancient = 0
def merge(a,b):
    global ancient
    x,y = find(a),find(b)
    if x == y:
        return True
    else:
        if x in ancient:
            parent[y] = x
            ancient -= {y}
        elif y in ancient:
            parent[x] = y
        else:
            ancient.add(x)
            parent[y] = x
        return False
while True:
    try:
        edges = []
        heapq.heapify(edges)
        for i in range((n:=int(input()))):
            a = list(map(int,input().split()))
            for j in range(i+1,n):
                heapq.heappush(edges,[a[j],i,j])
        parent = [i for i in range(n)]
        count,ancient = 0,set()
        while edges:
            x = heapq.heappop(edges)
            a,b = x[1],x[2]
            if not merge(a, b):
                count += x[0]
        print(count)
    except:
        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)

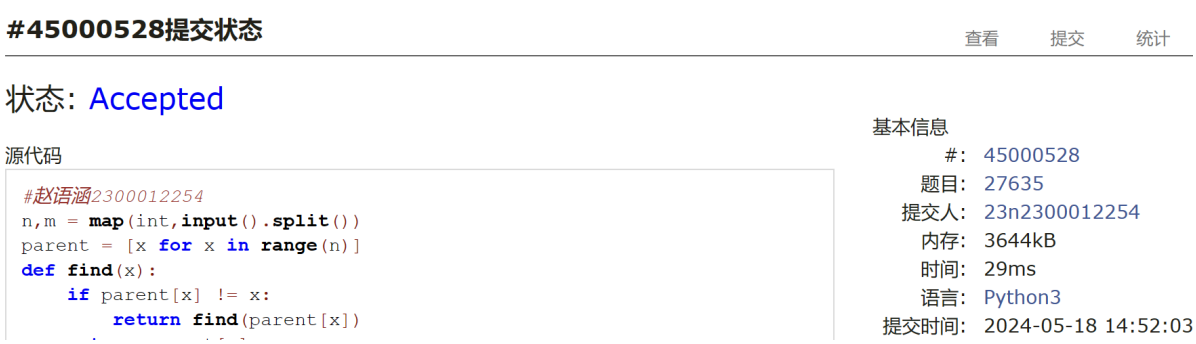
http://cs101.openjudge.cn/practice/27635/

思路：比较简单的并查集方法，直接将祖先中大的应该指向小的，如果在过程中有遇到某边的两个连接点是指向同一祖先的说明出现了连通情况loop=yes，最后统计根节点的数量如果只有1个根节点说明只有一个回路connected=yes

代码

```python
#赵语涵2300012254
n,m = map(int,input().split())
parent = [x for x in range(n)]
def find(x):
    if parent[x] != x:
        return find(parent[x])
    return parent[x]
loop = 'no'
for _ in range(m):
    a,b = map(int,input().split())
    x,y = find(a),find(b)
    if x == y:
        loop = 'yes'
    else:
        if x < y:
            parent[y] = x
        else:
            parent[x] = y
ancient = set(find(x) for x in range(n))
if len(ancient)==1:
    print('connected:yes')
else:
    print('connected:no')
print(f'loop:{loop}')
```

代码运行截图



#45000528提交状态

查看　提交　统计

状态: Accepted

源代码

```
#赵语涵2300012254
n,m = map(int,input().split())
parent = [x for x in range(n)]
def find(x):
    if parent[x] != x:
        return find(parent[x])
```

基本信息

#: 45000528
题目: 27635
提交人: 23n2300012254
内存: 3644kB
时间: 29ms
语言: Python3
提交时间: 2024-05-18 14:52:03

27947: 动态中位数

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

思路：因为看到了提示用heap，感觉是从heap特别的排列方法完成，但是对heap里面具体列表的放置顺序又不太清楚花了很多时间。然而并没有想到题解所示的维护2个不同的大小堆得到中位数。

代码

```python
import heapq

def dynamic_median(nums):
    # 维护小根和大根堆（对顶），保持中位数在大根堆的顶部
    min_heap = []  # 存储较大的一半元素，使用最小堆
    max_heap = []  # 存储较小的一半元素，使用最大堆

    median = []
    for i, num in enumerate(nums):
        # 根据当前元素的大小将其插入到对应的堆中
        if not max_heap or num <= -max_heap[0]:
            heapq.heappush(max_heap, -num)
        else:
            heapq.heappush(min_heap, num)

        # 调整两个堆的大小差，使其不超过 1
        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:
            median.append(-max_heap[0])

    return median

T = int(input())
for _ in range(T):
    #M = int(input())
    nums = list(map(int, input().split()))
    median = dynamic_median(nums)
    print(len(median))
    print(*median)
```

代码运行截图

#45191120提交状态 查看 提交 统计

状态: Accepted

源代码

```
import heapq

def dynamic_median(nums):
    # 维护小根和大根堆（对顶），保持中位数在大根堆的顶部
    min_heap = []  # 存储较大的一半元素，使用最小堆
    max_heap = []  # 存储较小的一半元素，使用最大堆
```

基本信息

#: 45191120
题目: 27947
提交人: 23n2300012254
内存: 10124kB
时间: 289ms
语言: Python3
提交时间: 2024-06-03 19:36:45

28190: 奶牛排队

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

思路：学习了单调栈的使用

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)

stack = []  # 清空栈以供寻找右边界使用

# 求右侧第一个≤h[i]的奶牛位
for i in range(N-1, -1, -1):
    while stack and heights[stack[-1]] > heights[i]:
        stack.pop()

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

    stack.append(i)

ans = 0
```

```python

# for i in range(N-1, -1, -1):  # 从大到小枚举是个技巧
#     for j in range(left_bound[i] + 1, i):
#         if right_bound[j] > i:
#             ans = max(ans, i - j + 1)
#             break
#
#     if i <= ans:
#         break

for i in range(N):  # 枚举右端点 B寻找 A，更新 ans
    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

#45193849提交状态

查看　提交　统计

状态: Accepted

源代码

```python
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
```

基本信息

#: 45193849
题目: 28190
提交人: 23n2300012254
内存: 92164kB
时间: 2817ms
语言: Python3
提交时间: 2024-06-04 01:08:42

2. 学习总结和收获

按照考试时间完成的话，只有AC4，感觉考试的时候这个难度要AC4也要看情况，如果有比较简单的题莫名其妙卡住思路或者思路一下没想对就很容易翻车（例如上学期计概）。总之尽量加强模板题学习吧，以及一些经典题例的总结和学习。看题解学到挺多东西。