

Assignment #D: May月考

Updated 2013 GMT+8 May 8, 2024

2024 spring, Compiled by 钟明衡 物理学院

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_NT x64 10.0.19045

Python编程环境: Visual Studio Code 1.76.1

C/C++编程环境: Visual Studio Code 1.76.1

1. 题目

02808: 校门外的树

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

思路:

经典的模拟题，直接把树标记即可

代码

```

1 ans, n = map(int, input().split())
2 ans += 1
3 trees = []
4 for i in range(0, ans+1):
5     trees.append(True)
6 for i in range(0, n):
7     start, stop = map(int, input().split())
8     for j in range(start, stop+1):
9         if trees[j]:
10             trees[j] = False
11             ans -= 1
12 print(ans)
13

```

代码运行截图

#41251237提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

ans, n = map(int, input().split())
ans += 1
trees = []
for i in range(0, ans+1):
    trees.append(True)
for i in range(0, n):
    start, stop = map(int, input().split())
    for j in range(start, stop+1):
        if trees[j]:
            trees[j] = False
            ans -= 1
print(ans)

```

基本信息

#: 41251237

题目: 02808

提交人: 23n2300011505(12号娱乐选手)

内存: 3900kB

时间: 46ms

语言: Python3

提交时间: 2023-09-19 15:56:20

20449: 是否被5整除

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

思路:

把数一位位用二进制补上并且判断即可

代码

```

1 s = input()
2 count = 0
3 for a in s:
4     count = (count << 1)+int(a)
5     if count % 5 == 0:
6         print(1, end='')
7     else:
8         print(0, end='')
9

```

代码运行截图

#44317206提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

s = input()
count = 0
for a in s:
    count = (count << 1)+int(a)
    if count % 5 == 0:
        print(1, end='')
    else:
        print(0, end='')

```

基本信息

#: 44317206

题目: 20449

提交人: 23n2300011505(12号娱乐选手)

内存: 3580kB

时间: 23ms

语言: Python3

提交时间: 2024-03-20 20:12:37

01258: Agri-Net

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

思路:

最小生成树，把边按照权重从小到大排序，每次用并查集判断一条边是否有必要添加

代码

```

1 p = []
2
3
4 def p(x):
5     if p[x] != x:
6         p[x] = p(p[x])
7     return p[x]
8
9
10 while True:
11     try:
12         n = int(input())
13     except EOFError:
14         break

```

```

15     ans = 0
16     M = [list(map(int, input().split())) for _ in range(n)]
17     p = [i for i in range(n)]
18     l = []
19     for i in range(n):
20         for j in range(n):
21             if i != j:
22                 l.append((i, j, M[i][j]))
23     l.sort(key=lambda x: x[2])
24     for i, j, k in l:
25         pi, pj = P(i), P(j)
26         if pi != pj:
27             p[pi] = pj
28             ans += k
29     print(ans)
30

```

代码运行截图

#44902134提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

p = []

def P(x):
    if p[x] != x:
        p[x] = P(p[x])
    return p[x]

while True:
    try:
        n = int(input())
    except EOFError:
        break
    ans = 0
    M = [list(map(int, input().split())) for _ in range(n)]
    p = [i for i in range(n)]
    l = []
    for i in range(n):
        for j in range(n):
            if i != j:
                l.append((i, j, M[i][j]))
    l.sort(key=lambda x: x[2])
    for i, j, k in l:
        pi, pj = P(i), P(j)
        if pi != pj:
            p[pi] = pj
            ans += k
    print(ans)

```

基本信息

#: 44902134

题目: 01258

提交人: 23n2300011505(12号娱乐选手)

内存: 4916kB

时间: 56ms

语言: Python3

提交时间: 2024-05-08 19:46:17

27635: 判断无向图是否连通有无回路(同23163)

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

思路:

用广度优先搜索来标记同一个连通区域, 如果标记完了一个连通区域, 发现还有点未标记, 则说明不连通

判断有无回路, 我的方法和之前笔试里的不一样, 我的方法是计算一个连通域中总边数和总节点数, 如果边数不小于节点数, 则存在环

代码

```
1 m, n = map(int, input().split())
2 l = []
3 g = [set() for _ in range(m)]
4 f = [0]*m
5 s = []
6 for _ in range(n):
7     a, b = map(int, input().split())
8     g[a].add(b)
9     g[b].add(a)
10    l.append(a)
11 s = set()
12 a, b = 'yes', 'no'
13 for i in range(m):
14     if not f[i]:
15         f[i] = 1
16         if s:
17             a = 'no'
18         s = set([i])
19         L = [i]
20         s, e = 0, 1
21         while s != e:
22             for j in range(s, e):
23                 for k in g[L[j]]:
24                     if not f[k]:
25                         f[k] = 1
26                         L.append(k)
27                         s.add(k)
28             s, e = e, len(L)
29         c = sum(j in s for j in l)
30         if c >= len(s):
31             b = 'yes'
32 print('connected:%s\nloop:%s' % (a, b))
33
```

代码运行截图

状态: Accepted

源代码

```
m, n = map(int, input().split())
l = []
g = [set() for _ in range(m)]
f = [0]*m
S = []
for _ in range(n):
    a, b = map(int, input().split())
    g[a].add(b)
    g[b].add(a)
    l.append(a)
S = set()
a, b = 'yes', 'no'
for i in range(m):
    if not f[i]:
        f[i] = 1
        if S:
            a = 'no'
        S = set([i])
        L = [i]
        s, e = 0, 1
        while s != e:
            for j in range(s, e):
                for k in g[L[j]]:
                    if not f[k]:
                        f[k] = 1
                        L.append(k)
                        S.add(k)
            s, e = e, len(L)
        c = sum(j in S for j in l)
        if c >= len(S):
            b = 'yes'
print('connected:%s\nloop:%s' % (a, b))
```

基本信息

#: 44672498
题目: 27635
提交人: 23n2300011505(12号娱乐选手)
内存: 4256kB
时间: 34ms
语言: Python3
提交时间: 2024-04-16 13:36:37

27947: 动态中位数

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

思路:

注意到每次中位数改变时，都是变为了前一个或者后一个，用两个堆正好能实现这个功能

比当前中位数小的，取负数放入small，大的则放入big，它们的堆顶始终是中位数的前一个或者后一个，另外用一个数来记录两边的非平衡，当计数达到2，就把多的那边堆顶的数取出成为新中位数，原来的中位数放入另一个堆，这样就不会超时了

代码

```
1 from heapq import heappush, heappop
2
3 for _ in range(int(input())):
4     l = list(map(int, input().split()))
5     small, big, a = [], [], l[0]
6     r = 0
7     mid = l[0]
```

```

8     c = 1
9     for x in l[1:]:
10         c += 1
11         if x >= mid:
12             heappush(big, x)
13             r += 1
14         else:
15             heappush(small, -x)
16             r -= 1
17         if c % 2:
18             if r == 2:
19                 heappush(small, -mid)
20                 mid = heappop(big)
21                 r = 0
22             if r == -2:
23                 heappush(big, mid)
24                 mid = -heappop(small)
25                 r = 0
26         a.append(mid)
27     print(len(a))
28     print(' '.join(map(str, a)))
29

```

代码运行截图

#44902355提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from heapq import heappush, heappop

for _ in range(int(input())):
    l = list(map(int, input().split()))
    small, big, a = [], [], [l[0]]
    r = 0
    mid = l[0]
    c = 1
    for x in l[1:]:
        c += 1
        if x >= mid:
            heappush(big, x)
            r += 1
        else:
            heappush(small, -x)
            r -= 1
        if c % 2:
            if r == 2:
                heappush(small, -mid)
                mid = heappop(big)
                r = 0
            if r == -2:
                heappush(big, mid)
                mid = -heappop(small)
                r = 0
        a.append(mid)
    print(len(a))
    print(' '.join(map(str, a)))

```

基本信息

#: 44902355

题目: 27947

提交人: 23n2300011505(12号娱乐选手)

内存: 10644kB

时间: 214ms

语言: Python3

提交时间: 2024-05-08 19:59:25

28190: 奶牛排队

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

思路：

N 只有 10^4 ，用 $O(N^2)$ 的算法是能过的

用两个指针搜索头尾，同时用 s 和 b 分别记录当前队列中最小和最大的值，当右侧加入的值不大于最左侧，左侧指针右移并重新开始搜索，而当最右侧的值是唯一的最大值，就和当前答案取更大者作为新的答案

代码

```
1 n = int(input())
2 l = [int(input()) for _ in range(n)]
3 s, b = -1, -1
4 ans = 0
5 for i in range(n-1):
6     s = b = l[i]
7     for j in range(i+1, n):
8         if l[j] <= s:
9             break
10        if l[j] > b:
11            b = l[j]
12            ans = max(ans, j-i+1)
13 print(ans)
14
```

代码运行截图

#44902469提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n = int(input())
l = [int(input()) for _ in range(n)]
s, b = -1, -1
ans = 0
for i in range(n-1):
    s = b = l[i]
    for j in range(i+1, n):
        if l[j] <= s:
            break
        if l[j] > b:
            b = l[j]
            ans = max(ans, j-i+1)
print(ans)
```

基本信息

#: 44902469

题目: 28190

提交人: 23n2300011505(12号娱乐选手)

内存: 3936kB

时间: 47ms

语言: Python3

提交时间: 2024-05-08 20:04:39

2. 学习总结和收获

月考挺难的，要不是有几题做过，真做不完

难题感觉全是计概风格，果然还是计概更难