Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by ==同学的姓名、院系==

罗瑞哲 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

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：

代码

```
#
L, m = map(int, input().split())
dp = [1]*(L+1)
for i in range(m):
    s, e = map(int, input().split())
    for j in range(s, e+1):
        dp[j] = 0
print(dp.count(1))
```

代码运行截图 ==（至少包含有"Accepted"）==

#45011716提交状态

状态: Accepted

源代码

```
L, m = map(int, input().split())
dp = [1]*(L+1)
for i in range(m):
    s, e = map(int, input().split())
    for j in range(s, e+1):
        dp[j] = 0
print(dp.count(1))
```

20449: 是否被5整除

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

思路:

代码

```
#
def binary_divisible_by_five(binary_string):
    result = ''
    num = 0
    for bit in binary_string:
        num = (num * 2 + int(bit)) % 5
        if num == 0:
            result += '1'
        else:
            result += '0'
    return result
binary_string = input().strip()
print(binary_divisible_by_five(binary_string))
```

代码运行截图 ==（至少包含有"Accepted"）==

#45011459提交状态

状态: Accepted

源代码

```
def binary_divisible_by_five(binary_string):
    result = ''
    num = 0
    for bit in binary_string:
        num = (num * 2 + int(bit)) % 5
        if num == 0:
            result += '1'
        else:
            result += '0'
    return result
binary_string = input().strip()
print(binary_divisible_by_five(binary_string))
```

01258: Agri-Net

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

思路:

代码

```
#
from heapq import heappop, heappush
while True:
    try:
        n = int(input())
    except:
        break
    mat, cur = [], 0
    for i in range(n):
        mat.append(list(map(int, input().split())))
    d, v, q, cnt = [100000 for i in range(n)], set(), [], 0
    d[0] = 0
    heappush(q, (d[0], 0))
    while q:
        x, y = heappop(q)
        if y in v:
            continue
        v.add(y)
        cnt += d[y]
        for i in range(n):
            if d[i] > mat[y][i]:
                d[i] = mat[y][i]
                heappush(q, (d[i], i))
```

```
    print(cnt)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#45011543提交状态

状态: Accepted

源代码

```
from heapq import heappop, heappush
while True:
    try:
        n = int(input())
    except:
        break
    mat, cur = [], 0
    for i in range(n):
        mat.append(list(map(int, input().split())))
    d, v, q, cnt = [100000 for i in range(n)], set(), [], 0
    d[0] = 0
    heappush(q, (d[0], 0))
    while q:
        x, y = heappop(q)
        if y in v:
            continue
        v.add(y)
        cnt += d[y]
        for i in range(n):
            if d[i] > mat[y][i]:
                d[i] = mat[y][i]
                heappush(q, (d[i], i))
    print(cnt)
```

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

```
#
n, m = list(map(int, input().split()))
edge = [[]for _ in range(n)]
for _ in range(m):
    a, b = list(map(int, input().split()))
    edge[a].append(b)
    edge[b].append(a)
cnt, flag = set(), False
```

```
def dfs(x, y):
    global cnt, flag
    cnt.add(x)
    for i in edge[x]:
        if i not in cnt:
            dfs(i, x)
        elif y != i:
            flag = True
for i in range(n):
    cnt.clear()
    dfs(i, -1)
    if len(cnt) == n:
        break
    if flag:
        break
print("connected:"+("yes" if len(cnt) == n else "no"))
print("loop:"+("yes" if flag else 'no'))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#45011578提交状态

状态: Accepted

源代码

```
n, m = list(map(int, input().split()))
edge = [[]for _ in range(n)]
for _ in range(m):
    a, b = list(map(int, input().split()))
    edge[a].append(b)
    edge[b].append(a)
cnt, flag = set(), False
def dfs(x, y):
    global cnt, flag
    cnt.add(x)
    for i in edge[x]:
        if i not in cnt:
            dfs(i, x)
        elif y != i:
            flag = True
for i in range(n):
    cnt.clear()
    dfs(i, -1)
    if len(cnt) == n:
        break
    if flag:
        break
print("connected:"+("yes" if len(cnt) == n else "no"))
print("loop:"+("yes" if flag else 'no'))
```

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 = []
    median = []
    for i, num in enumerate(nums):
        if not max_heap 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 % 2 == 0:
            median.append(-max_heap[0])
    return median
T = int(input())
for _ in range(T):
    nums = list(map(int, input().split()))
    median = dynamic_median(nums)
    print(len(median))
    print(*median)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#45011618提交状态

状态: Accepted

源代码

```
import heapq
def dynamic_median(nums):
    min_heap = []
    max_heap = []
    median = []
    for i, num in enumerate(nums):
        if not max_heap 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 % 2 == 0:
            median.append(-max_heap[0])
    return median
T = int(input())
for _ in range(T):
    nums = list(map(int, input().split()))
    median = dynamic_median(nums)
    print(len(median))
    print(*median)
```

28190: 奶牛排队

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

思路:

代码

```
#
N = int(input())
heights = [int(input()) for _ in range(N)]
left_bound = [-1] * N
right_bound = [N] * N
stack = []
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 = []
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
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"）==

#45011665提交状态

状态: Accepted

源代码

```
N = int(input())
heights = [int(input()) for _ in range(N)]
left_bound = [-1] * N
right_bound = [N] * N
stack = []
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 = []
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
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)
```

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。==

临近期末，事也越来越多了，我得抓紧了。