Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by ==同学的姓名、院系==

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/

代码

```
n=[int(i) for i in input().split()]
i=0
l=[]
while i<=n[1]-1:
    k=[int(j) for j in input().split()]
    l.append(k)
    i+=1
w=[i for i in range(n[0]+1)]
a=[]
for t in l:
    a+=w[t[0]:t[1]+1]
o=set(a)
x=len(w)-len(o)
print(x)
##以前写过
```

#44960744提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=[int(i) for i in input().split()]
i=0
l=[]
while i<=n[1]-1:
    k=[int(j) for j in input().split()]
    l.append(k)
    i+=1
w=[i for i in range(n[0]+1)]
a=[]
for t in l:
    a+=w[t[0]:t[1]+1]
o=set(a)
x=len(w)-len(o)
print(x)
```

基本信息

#: 44960744
题目: 02808
提交人: 22n2200011800
内存: 7428kB
时间: 28ms
语言: Python3
提交时间: 2024-05-14 15:47:03

English 帮助 关于

20449: 是否被5整除

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

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))
##遍历直接模拟即可
```

#44960765提交状态

查看 提交 统计 提问

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))
```

基本信息

#: 44960765
题目: 20449
提交人: 22n2200011800
内存: 3616kB
时间: 22ms
语言: Python3
提交时间: 2024-05-14 15:48:31

English 帮助 关于

01258: Agri-Net

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

代码

```
from heapq import heappop, heappush, heapify

def prim(graph, start_node):
    mst = set()
    visited = set([start_node])
    edges = [
        (cost, start_node, to)
        for to, cost in graph[start_node].items()
    ]
    heapify(edges)

    while edges:
        cost, frm, to = heappop(edges)
        if to not in visited:
            visited.add(to)
            mst.add((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in visited:
                    heappush(edges, (cost2, to, to_next))

    return mst


while True:
    try:
        N = int(input())
    except EOFError:
        break

    graph = {i: {} for i in range(N)}
    for i in range(N):
        for j, cost in enumerate(map(int, input().split())):
            graph[i][j] = cost

    mst = prim(graph, 0)
    total_cost = sum(cost for frm, to, cost in mst)
    print(total_cost)
##用Prim算法可以解决
```

#44960823提交状态 查看 提交 统计 提问

状态: Accepted

基本信息
#: 44960823
题目: 01258
提交人: 22n2200011800
内存: 4832kB
时间: 43ms
语言: Python3
提交时间: 2024-05-14 15:51:20

源代码

```
from heapq import heappop, heappush, heapify

def prim(graph, start_node):
    mst = set()
    visited = set([start_node])
    edges = [
        (cost, start_node, to)
        for to, cost in graph[start_node].items()
    ]
    heapify(edges)

    while edges:
        cost, frm, to = heappop(edges)
        if to not in visited:
            visited.add(to)
            mst.add((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in visited:
                    heappush(edges, (cost2, to, to_next))

    return mst


while True:
    try:
        N = int(input())
    except EOFError:
        break

    graph = {i: {} for i in range(N)}
    for i in range(N):
        for j, cost in enumerate(map(int, input().split())):
            graph[i][j] = cost

    mst = prim(graph, 0)
    total_cost = sum(cost for frm, to, cost in mst)
    print(total_cost)
```

27635: 判断无向图是否连通有无回路(同23163)

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

代码

```
def is_connected(graph, n):
    visited = [False] * n
    stack = [0]
    visited[0] = True

    while stack:
        node = stack.pop()
        for neighbor in graph[node]:
            if not visited[neighbor]:
                stack.append(neighbor)
                visited[neighbor] = True

    return all(visited)

def has_cycle(graph, n):
    def dfs(node, visited, parent):
        visited[node] = True
        for neighbor in graph[node]:
            if not visited[neighbor]:
                if dfs(neighbor, visited, node):
                    return True
            elif parent != neighbor:
                return True
        return False
```

```
    visited = [False] * n
    for node in range(n):
        if not visited[node]:
            if dfs(node, visited, -1):
                return True
    return False

n, m = map(int, input().split())
graph = [[] for _ in range(n)]
for _ in range(m):
    u, v = map(int, input().split())
    graph[u].append(v)
    graph[v].append(u)

connected = is_connected(graph, n)
has_loop = has_cycle(graph, n)
print("connected:yes" if connected else "connected:no")
print("loop:yes" if has_loop else "loop:no")
##dfs的运用
```

#44960888提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 44960888
题目: 27635
提交人: 22n2200011800
内存: 3700kB
时间: 26ms
语言: Python3
提交时间: 2024-05-14 15:55:21

源代码

```
def is_connected(graph, n):
    visited = [False] * n
    stack = [0]
    visited[0] = True

    while stack:
        node = stack.pop()
        for neighbor in graph[node]:
            if not visited[neighbor]:
                stack.append(neighbor)
                visited[neighbor] = True

    return all(visited)

def has_cycle(graph, n):
    def dfs(node, visited, parent):
        visited[node] = True
        for neighbor in graph[node]:
            if not visited[neighbor]:
                if dfs(neighbor, visited, node):
                    return True
            elif parent != neighbor:
                return True
        return False

    visited = [False] * n
    for node in range(n):
        if not visited[node]:
            if dfs(node, visited, -1):
                return True
    return False

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

27947: 动态中位数

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

代码

```
import heapq
```

```python
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)
##考试的时间不够了，后面下来学习了题解
```

#44960909提交状态

查看 提交 统计 提问

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)
```

基本信息

#: 44960909
题目: 27947
提交人: 22n2200011800
内存: 10900kB
时间: 289ms
语言: Python3
提交时间: 2024-05-14 15:56:53

English 帮助 关于

28190: 奶牛排队

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

代码

```
from bisect import bisect_right as bl
lis,q1,q2,ans=[int(input())for _ in range(int(input()))],[-1],[-1],0
for i in range(len(lis)):
    while len(q1)>1 and lis[q1[-1]]>=lis[i]:q1.pop()
    while len(q2)>1 and lis[q2[-1]]<lis[i]:q2.pop()
    id=bl(q1,q2[-1])
    if id<len(q1):ans=max(ans,i-q1[id]+1)
    q1.append(i)
    q2.append(i)
print(ans)
##学习了同学的解法，简洁优美
```

#44960988提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from bisect import bisect_right as bl
lis,q1,q2,ans=[int(input())for _ in range(int(input()))],[-1],[-1],0
for i in range(len(lis)):
    while len(q1)>1 and lis[q1[-1]]>=lis[i]:q1.pop()
    while len(q2)>1 and lis[q2[-1]]<lis[i]:q2.pop()
    id=bl(q1,q2[-1])
    if id<len(q1):ans=max(ans,i-q1[id]+1)
    q1.append(i)
    q2.append(i)
print(ans)
```

基本信息

#:	44960988
题目:	28190
提交人:	22n2200011800
内存:	40280kB
时间:	2258ms
语言:	Python3
提交时间:	2024-05-14 16:00:01

2. 学习总结和收获

这次正常考试时间内大概能ac4的样子，还是不够，继续加油