Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by 周添 物理学院

1. 题目

02808: 校门外的树

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

代码

```
a, b = map(int, input().split())
c = [1]

for ii in range(1, a+1):
    c.append(1)
for i in range(b):
    m, n = map(int, input().split())
    for j in range(m, n+1):
        c[j] = 0

nu = 0

for ij in c:
    if int(ij):
        nu += 1

print(nu)
```

状态: Accepted

源代码

```
a, b = map(int, input().split())
c = [1]

for ii in range(1, a+1):
    c.append(1)
for i in range(b):
    m, n = map(int, input().split())
    for j in range(m, n+1):
        c[j] = 0

nu = 0

for ij in c:
    if int(ij):
        nu += 1

print(nu)
```

基本信息

#: 41249418
题目: 02808
提交人: 23n2300011538
内存: 3676kB
时间: 50ms
语言: Python3
提交时间: 2023-09-19 13:02:22

20449: 是否被5整除

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

代码

```
def divisible_by_5(A):
    result = []
    prefix = 0
    for bit in A:
        prefix = (prefix * 2 + int(bit)) % 5
        result.append(1 if prefix == 0 else 0)
    return ''.join(map(str, result))


input_string = input()
print(divisible_by_5(input_string))
```

状态: Accepted

源代码

```
def divisible_by_5(A):
    result = []
    prefix = 0
    for bit in A:
        prefix = (prefix * 2 + int(bit)) % 5
        result.append(1 if prefix == 0 else 0)
    return ''.join(map(str, result))


input_string = input()
print(divisible_by_5(input_string))
```

基本信息

#:	44934995
题目:	20449
提交人:	23n2300011538
内存:	3940kB
时间:	24ms
语言:	Python3
提交时间:	2024-05-12 09:09:13

01258: Agri-Net

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

代码

```
def prim(n, e):
    value_sum = 0
    dis = []
    book = [0] * n

    for i in range(0, n):
        dis.append(e[0][i])

    dis[0] = 0
    book[0] = 1

    for i in range(0, n-1):
        min_val = float('inf')
        for j in range(0, n):
            if book[j] == 0 and dis[j] < min_val:
                min_val = dis[j]
                # print(min_val)
```

```
                k = j

        value_sum += min_val
        # print(value_sum)
        book[k] = 1

        for j in range(0, n):
            if book[j] == 0 and dis[j] > e[k][j]:
                dis[j] = e[k][j]

    print(value_sum)


while True:
    try:
        n = int(input())
        e = []
        for i in range(n):
            e.append(list(map(int, input().split())))
        prim(n, e)
    except EOFError:
        break
```

状态: Accepted

源代码

```
def prim(n, e):
    value_sum = 0
    dis = []
    book = [0] * n

    for i in range(0, n):
        dis.append(e[0][i])

    dis[0] = 0
    book[0] = 1

    for i in range(0, n-1):
        min_val = float('inf')
        for j in range(0, n):
            if book[j] == 0 and dis[j] < min_val:
                min_val = dis[j]
                # print(min_val)
                k = j

        value_sum += min_val
        # print(value_sum)
        book[k] = 1

        for j in range(0, n):
            if book[j] == 0 and dis[j] > e[k][j]:
```

基本信息

#:	44982297
题目:	01258
提交人:	23n2300011538
内存:	3912kB
时间:	28ms
语言:	Python3
提交时间:	2024-05-16 19:13:57

27635: 判断无向图是否连通有无回路(同23163)

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

代码

```
n, m = map(int, input().split())
```

```
edges = []
visited_for_connection = [False]*n

for i in range(m):
    a, b = map(int, input().split())
    edges.append((a, b))
    visited_for_connection[a] = True
    visited_for_connection[b] = True

if False in visited_for_connection:
    print('connected:no')
else:
    print('connected:yes')

where_to_go = [0]*n


def loop_ever():
    k = 1
    for x, y in edges:
        # print(x, y)
        if where_to_go[x] == 0 and where_to_go[y] == 0:
            where_to_go[x] = k
            where_to_go[y] = k
            k += 1
        elif where_to_go[x] != 0 and where_to_go[y] == 0:
            where_to_go[y] = where_to_go[x]
        elif where_to_go[y] != 0 and where_to_go[x] == 0:
            where_to_go[x] = where_to_go[y]
        elif where_to_go[x] == where_to_go[y]:
            return True
        else:
            s = where_to_go[x]
            t = where_to_go[y]
            for i in range(n):
                if where_to_go[i] == s:
                    where_to_go[i] = t
    return False


if loop_ever():
    print('loop:yes')
else:
    print('loop:no')
```

状态: Accepted

源代码

```
n, m = map(int, input().split())
edges = []
visited_for_connection = [False]*n

for i in range(m):
    a, b = map(int, input().split())
    edges.append((a, b))
    visited_for_connection[a] = True
    visited_for_connection[b] = True

if False in visited_for_connection:
    print('connected:no')
else:
    print('connected:yes')

where_to_go = [0]*n


def loop_ever():
    k = 1
    for x, y in edges:
        # print(x, y)
        if where_to_go[x] == 0 and where_to_go[y] == 0:
            where_to_go[x] = k
```

基本信息

#: 44982914
题目: 27635
提交人: 23n2300011538
内存: 3820kB
时间: 29ms
语言: Python3
提交时间: 2024-05-16 19:42:14

27947: 动态中位数

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

代码

```python
import heapq


def median(nums):
    max_heap = []
    min_heap = []  # bigger

    result = []
    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:
            result.append(-max_heap[0])

    return result
```

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

状态: Accepted

源代码

```
import heapq


def median(nums):
    max_heap = []
    min_heap = []  # bigger

    result = []
    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:
            result.append(-max_heap[0])

    return result
```

基本信息

#:	44962767
题目:	27947
提交人:	23n2300011538
内存:	10864kB
时间:	296ms
语言:	Python3
提交时间:	2024-05-14 18:10:55

28190: 奶牛排队

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

代码

```
n = int(input())
a = []
for i in range(n):
    a.append(int(input()))

a.insert(0, float('inf'))
a.append(0)

Amin = [0] * (n + 1)
Bmax = [0] * (n + 2)

q = [0] * (n + 2)
num = 0

for i in range(1, n + 1):
    while num and a[q[num]] < a[i]:
        num -= 1
    Amin[i] = q[num] + 1
```

```
    num += 1
    q[num] = i

q = [0] * (n + 2)
q[0] = n + 1
num = 0

for i in range(n, 0, -1):
    while num and a[q[num]] > a[i]:
        num -= 1
    Bmax[i] = q[num] - 1
    num += 1
    q[num] = i

ans = 0

for i in range(n, 0, -1):
    if ans >= i:
        break
    for j in range(Amin[i], i):
        if Bmax[j] >= i:
            ans = max(ans, i - j + 1)
            break

print(ans)
```

状态: Accepted

源代码

```
n = int(input())
a = []
for i in range(n):
    a.append(int(input()))

a.insert(0, float('inf'))
a.append(0)

Amin = [0] * (n + 1)
Bmax = [0] * (n + 2)

q = [0] * (n + 2)
num = 0

for i in range(1, n + 1):
    while num and a[q[num]] < a[i]:
        num -= 1
    Amin[i] = q[num] + 1
    num += 1
    q[num] = i

q = [0] * (n + 2)
q[0] = n + 1
num = 0
```

基本信息

#:	44976268
题目:	28190
提交人:	23n2300011538
内存:	122632kB
时间:	2838ms
语言:	Python3
提交时间:	2024-05-15 23:15:40

2. 学习总结和收获

agri-net好不容易看完了题干没看输入输出要求，没看到会输入several组数据，直接原地薄壁。