Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by ==同学的姓名、院系==

赵云天 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：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/

思路：用集合的方法写

代码

```
l,m = map(int,input().split())
a=set([i for i in range(0,l+1)])
jian =set()
for j in range(m):
    x,y = map(int,input().split())
    b = set([k for k in range(x,y+1)])
    jian = jian | b
a = a - jian
a = sorted(list(a))
print(len(a))
```

代码运行截图 ==（至少包含有"Accepted"）==

20449: 是否被5整除

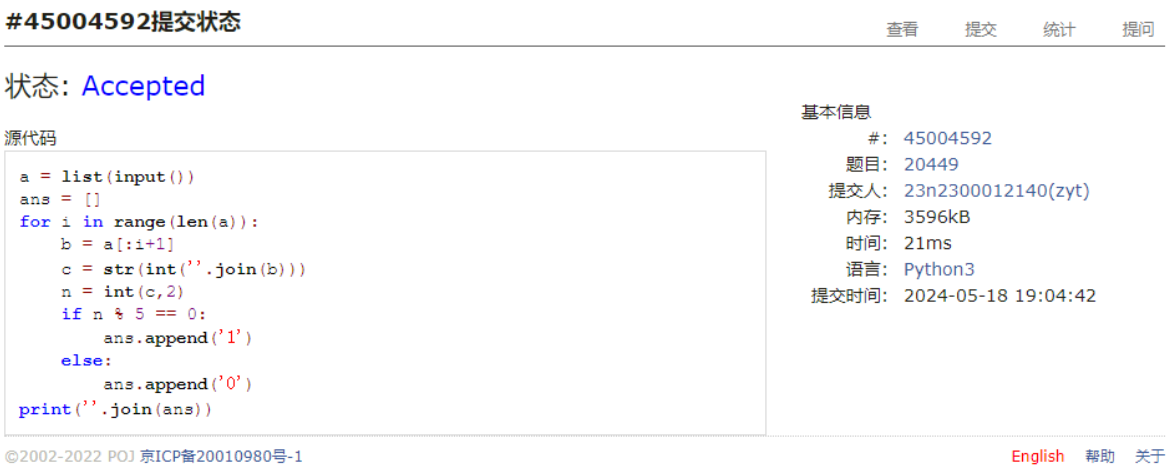
http://cs101.openjudge.cn/practice/20449/

思路：简单的二进制转换

代码

```
a = list(input())
ans = []
for i in range(len(a)):
    b = a[:i+1]
    c = str(int(''.join(b)))
    n = int(c,2)
    if n % 5 == 0:
        ans.append('1')
    else:
        ans.append('0')
print(''.join(ans))
```

代码运行截图 ==（至少包含有"Accepted"）==



#45004592提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
a = list(input())
ans = []
for i in range(len(a)):
    b = a[:i+1]
    c = str(int(''.join(b)))
    n = int(c,2)
    if n % 5 == 0:
        ans.append('1')
    else:
        ans.append('0')
print(''.join(ans))
```

基本信息

#: 45004592
题目: 20449
提交人: 23n2300012140(zyt)
内存: 3596kB
时间: 21ms
语言: Python3
提交时间: 2024-05-18 19:04:42

English 帮助 关于

01258: Agri-Net

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

思路：比较模板的题目

代码

```
class dis:
    def __init__(self,n):
        self.parent =list(range(n))
        self.rank = [0]*n

    def find(self,x):
        if self.parent[x] != x:
            self.parent[x] = self.find(self.parent[x])
        return self.parent[x]

    def pas(self,x,y):
        xp = self.find(x)
        yp = self.find(y)
        if xp == yp:
            return False
        else:
            if self.rank[xp] < self.rank[yp]:
                self.parent[xp] = yp
            if self.rank[xp] > self.rank[yp]:
                self.parent[yp] = xp
            else:
                self.parent[xp] = yp
                self.rank[yp] += 1
        return True

def kru(edge,n):
```

```
    a = dis(n)
    ans = 0
    for value,u,v in sorted(edge):
        if a.pas(u,v):
            ans += value
    return ans

while True:
    try:
        n = int(input())
        edge = []
        for i in range(n):
            a = list(map(int,input().split()))
            for j in range(i+1,n):
                edge.append((a[j],i,j))
        print(kru(edge,n))
    except EOFError:
        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：看的题解的方法

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")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#45038671提交状态

查看 提交 统计 提问

状态: Accepted

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:
```

基本信息

#: 45038671
题目: 27635
提交人: 23n2300012140(zyt)
内存: 3696kB
时间: 25ms
语言: Python3
提交时间: 2024-05-21 22:55:59

27947: 动态中位数

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

思路：看的题解的方法

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"）==

#45038814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))
```

基本信息

#: 45038814
题目: 27947
提交人: 23n2300012140(zyt)
内存: 10916kB
时间: 292ms
语言: Python3
提交时间: 2024-05-21 23:10:03

28190: 奶牛排队

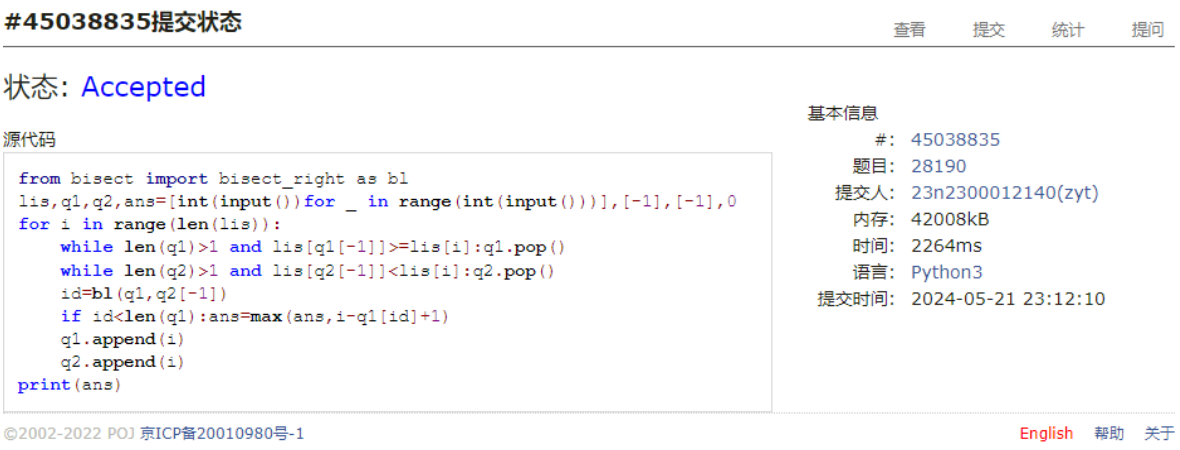
http://cs101.openjudge.cn/practice/28190/

思路：看的题解的方法

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)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#45038835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

#: 45038835
题目: 28190
提交人: 23n2300012140(zyt)
内存: 42008kB
时间: 2264ms
语言: Python3
提交时间: 2024-05-21 23:12:10

English 帮助 关于

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。==

前四题还可以，后面两题不好写