Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by Xinjie Song, Phy

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 11 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

02808: 校门外的树

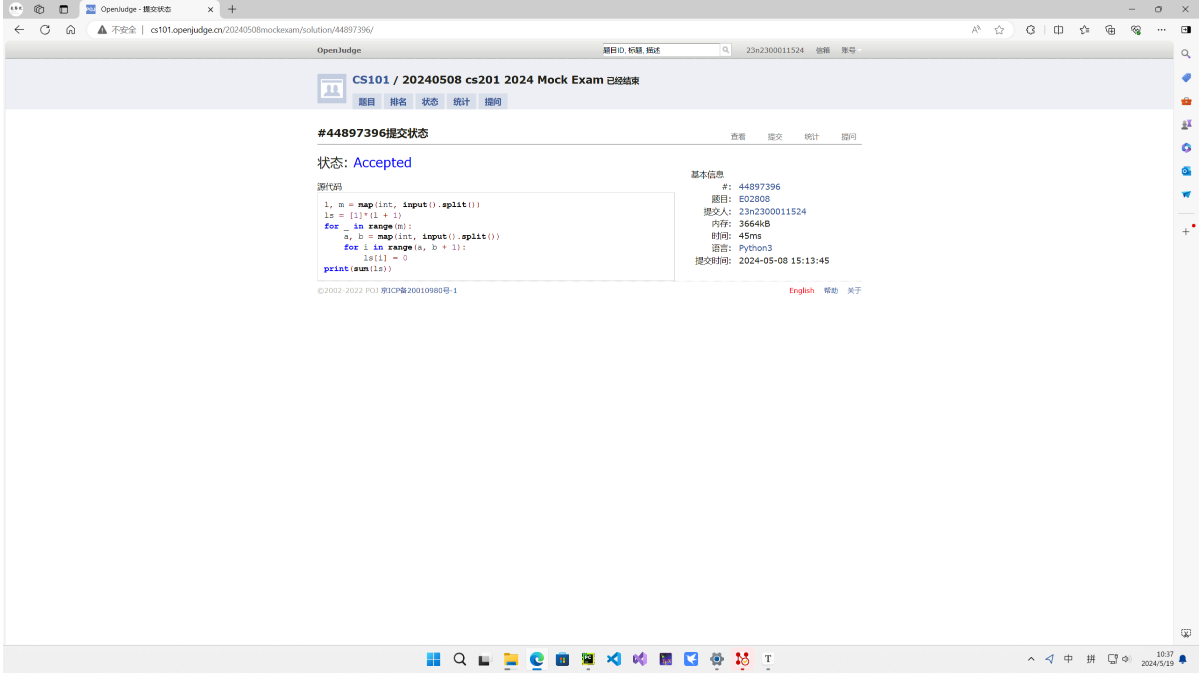
http://cs101.openjudge.cn/practice/02808/

思路：模拟法

代码

```
l, m = map(int, input().split())
ls = [1]*(l + 1)
for _ in range(m):
    a, b = map(int, input().split())
    for i in range(a, b + 1):
        ls[i] = 0
print(sum(ls))
```

代码运行截图



#44897396提交状态

状态: Accepted

源代码

```python
l, m = map(int, input().split())
ls = [1]*(l + 1)
for _ in range(m):
    a, b = map(int, input().split())
    for i in range(a, b + 1):
        ls[i] = 0
print(sum(ls))
```

基本信息

#:	44897396
题目:	E02808
提交人:	23n2300011524
内存:	3664kB
时间:	45ms
语言:	Python3
提交时间:	2024-05-08 15:13:45

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

20449: 是否被5整除

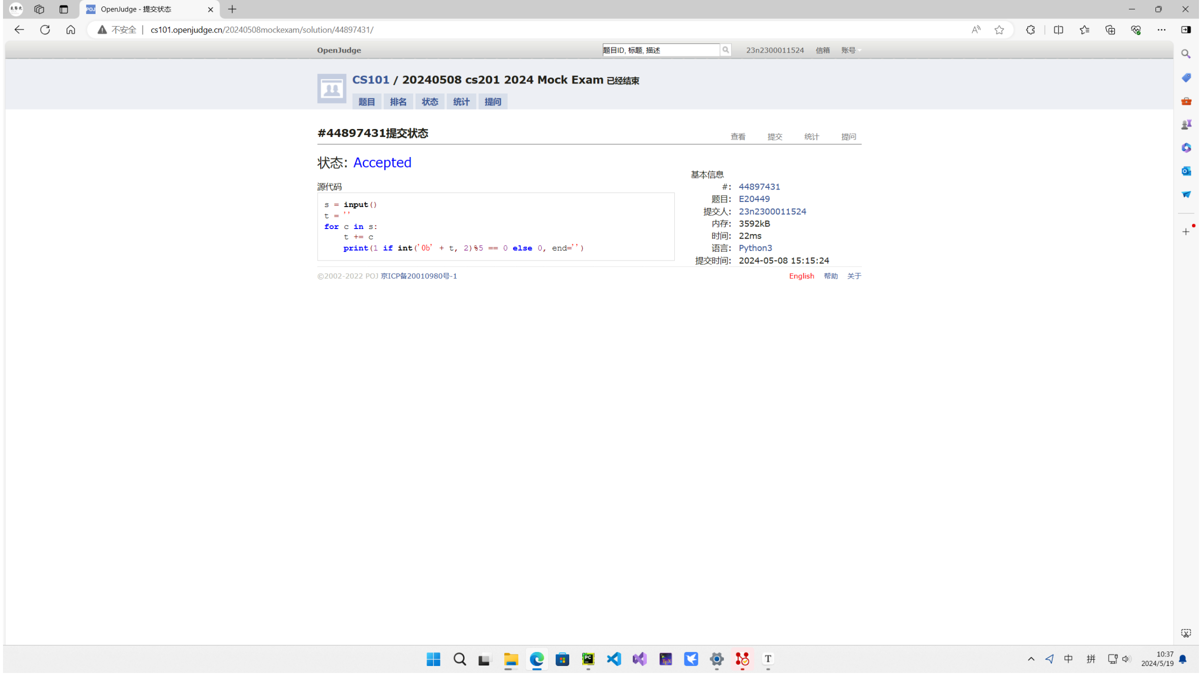
http://cs101.openjudge.cn/practice/20449/

思路：利用内置转换

代码

```python
s = input()
t = ''
for c in s:
    t += c
    print(1 if int('0b' + t, 2)%5 == 0 else 0, end='')
```

代码运行截图



OpenJudge

CS101 / 20240508 cs201 2024 Mock Exam 已经结束

#44897431提交状态

状态: Accepted

源代码

```
s = input()
t = ''
for c in s:
    t += c
    print(1 if int('0b' + t, 2)%5 == 0 else 0, end='')
```

基本信息

#:	44897431
题目:	E20449
提交人:	23n2300011524
内存:	3592kB
时间:	22ms
语言:	Python3
提交时间:	2024-05-08 15:15:24

©2002-2022 POJ 京ICP备20010980号-1

01258: Agri-Net

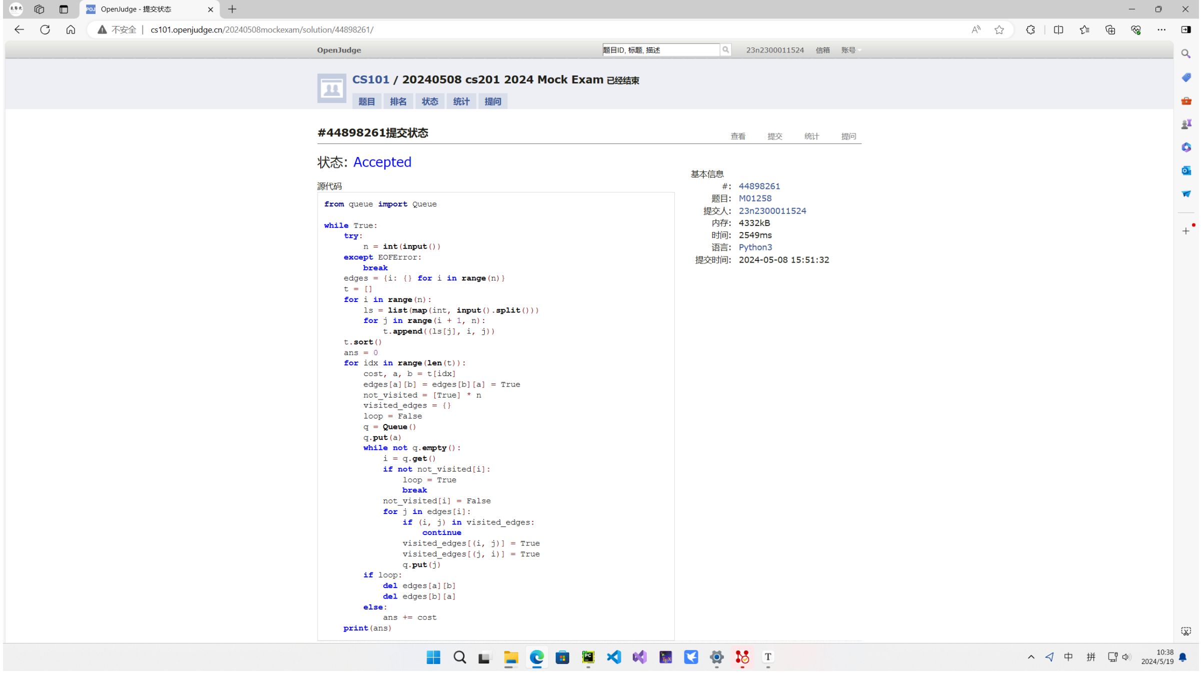
http://cs101.openjudge.cn/practice/01258/

思路：拓扑排序

代码

```
from queue import Queue

while True:
    try:
        n = int(input())
    except EOFError:
        break
    edges = {i: {} for i in range(n)}
    t = []
    for i in range(n):
        ls = list(map(int, input().split()))
        for j in range(i + 1, n):
            t.append((ls[j], i, j))
    t.sort()
    ans = 0
    for idx in range(len(t)):
        cost, a, b = t[idx]
        edges[a][b] = edges[b][a] = True
        not_visited = [True] * n
        visited_edges = {}
        loop = False
        q = Queue()
        q.put(a)
        while not q.empty():
            i = q.get()
            if not not_visited[i]:
                loop = True
                break
            not_visited[i] = False
            for j in edges[i]:
                if (i, j) in visited_edges:
                    continue
                visited_edges[(i, j)] = True
                visited_edges[(j, i)] = True
                q.put(j)
        if loop:
            del edges[a][b]
            del edges[b][a]
        else:
            ans += cost
    print(ans)
```

代码运行截图



CS101 / 20240508 cs201 2024 Mock Exam 已经结束

#44898261提交状态

状态: Accepted

源代码

```
from queue import Queue

while True:
    try:
        n = int(input())
    except EOFError:
        break
    edges = {i: {} for i in range(n)}
    t = []
    for i in range(n):
        ls = list(map(int, input().split()))
        for j in range(i + 1, n):
            t.append((ls[j], i, j))
    t.sort()
    ans = 0
    for idx in range(len(t)):
        cost, a, b = t[idx]
        edges[a][b] = edges[b][a] = True
        not_visited = [True] * n
        visited_edges = {}
        loop = False
        q = Queue()
        q.put(a)
        while not q.empty():
            i = q.get()
            if not not_visited[i]:
                loop = True
                break
            not_visited[i] = False
            for j in edges[i]:
                if (i, j) in visited_edges:
                    continue
                visited_edges[(i, j)] = True
                visited_edges[(j, i)] = True
                q.put(j)
        if loop:
            del edges[a][b]
            del edges[b][a]
        else:
            ans += cost
    print(ans)
```

基本信息

#:	44898261
题目:	M01258
提交人:	23n2300011524
内存:	4332kB
时间:	2549ms
语言:	Python3
提交时间:	2024-05-08 15:51:32

27635: 判断无向图是否连通有无回路(同23163)

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

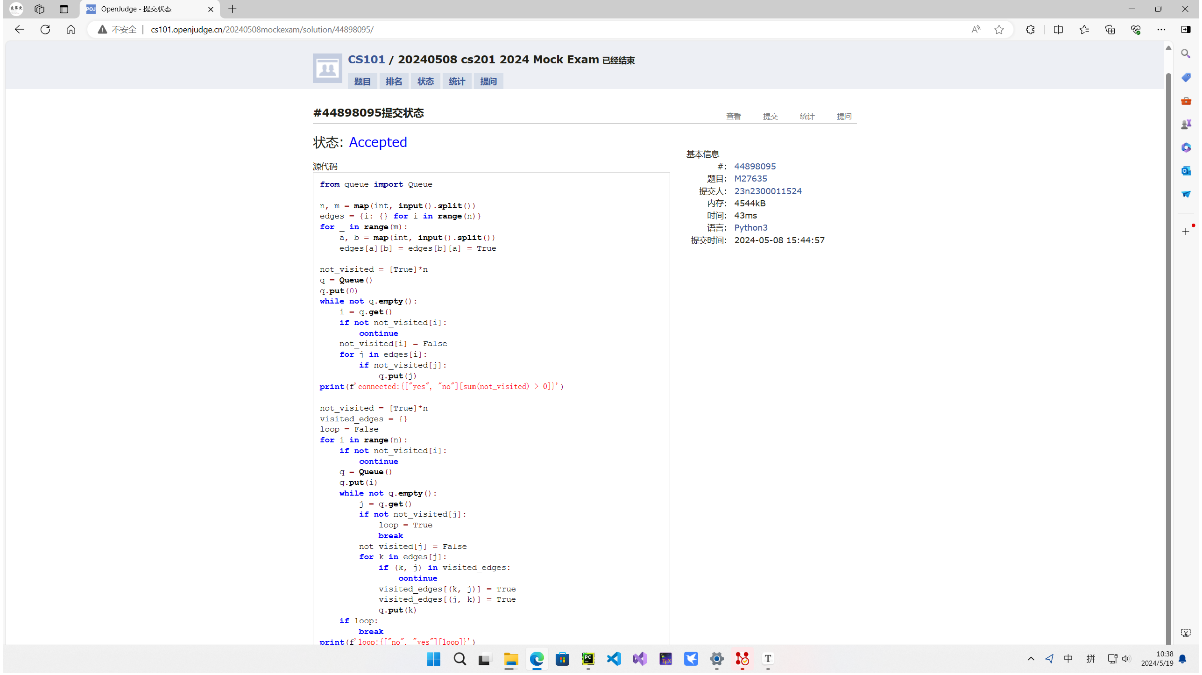
思路：常规思路

代码

```
from queue import Queue

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

```
edges = {i: {} for i in range(n)}
for _ in range(m):
    a, b = map(int, input().split())
    edges[a][b] = edges[b][a] = True

not_visited = [True]*n
q = Queue()
q.put(0)
while not q.empty():
    i = q.get()
    if not not_visited[i]:
        continue
    not_visited[i] = False
    for j in edges[i]:
        if not_visited[j]:
            q.put(j)
print(f'connected:{["yes", "no"][sum(not_visited) > 0]}')

not_visited = [True]*n
visited_edges = {}
loop = False
for i in range(n):
    if not not_visited[i]:
        continue
    q = Queue()
    q.put(i)
    while not q.empty():
        j = q.get()
        if not not_visited[j]:
            loop = True
            break
        not_visited[j] = False
        for k in edges[j]:
            if (k, j) in visited_edges:
                continue
            visited_edges[(k, j)] = True
            visited_edges[(j, k)] = True
            q.put(k)
    if loop:
        break
print(f'loop:{["no", "yes"][loop]}')
```

代码运行截图



CS101 / 20240508 cs201 2024 Mock Exam 已经结束

#44898095提交状态

状态: Accepted

源代码

```
from queue import Queue

n, m = map(int, input().split())
edges = {i: {} for i in range(n)}
for _ in range(m):
    a, b = map(int, input().split())
    edges[a][b] = edges[b][a] = True

not_visited = [True]*n
q = Queue()
q.put(0)
while not q.empty():
    i = q.get()
    if not not_visited[i]:
        continue
    not_visited[i] = False
    for j in edges[i]:
        if not_visited[j]:
            q.put(j)
print(f'connected:{["yes", "no"][sum(not_visited) > 0]}')

not_visited = [True]*n
visited_edges = {}
loop = False
for i in range(n):
    if not not_visited[i]:
        continue
    q = Queue()
    q.put(i)
    while not q.empty():
        j = q.get()
        if not not_visited[j]:
            loop = True
            break
        not_visited[j] = False
        for k in edges[j]:
            if (k, j) in visited_edges:
                continue
            visited_edges[(k, j)] = True
            visited_edges[(j, k)] = True
            q.put(k)
    if loop:
        break
print(f'loop:{["no", "yes"][loop]}')
```

基本信息

#: 44898095
题目: M27635
提交人: 23n2300011524
内存: 4544kB
时间: 43ms
语言: Python3
提交时间: 2024-05-08 15:44:57

27947: 动态中位数

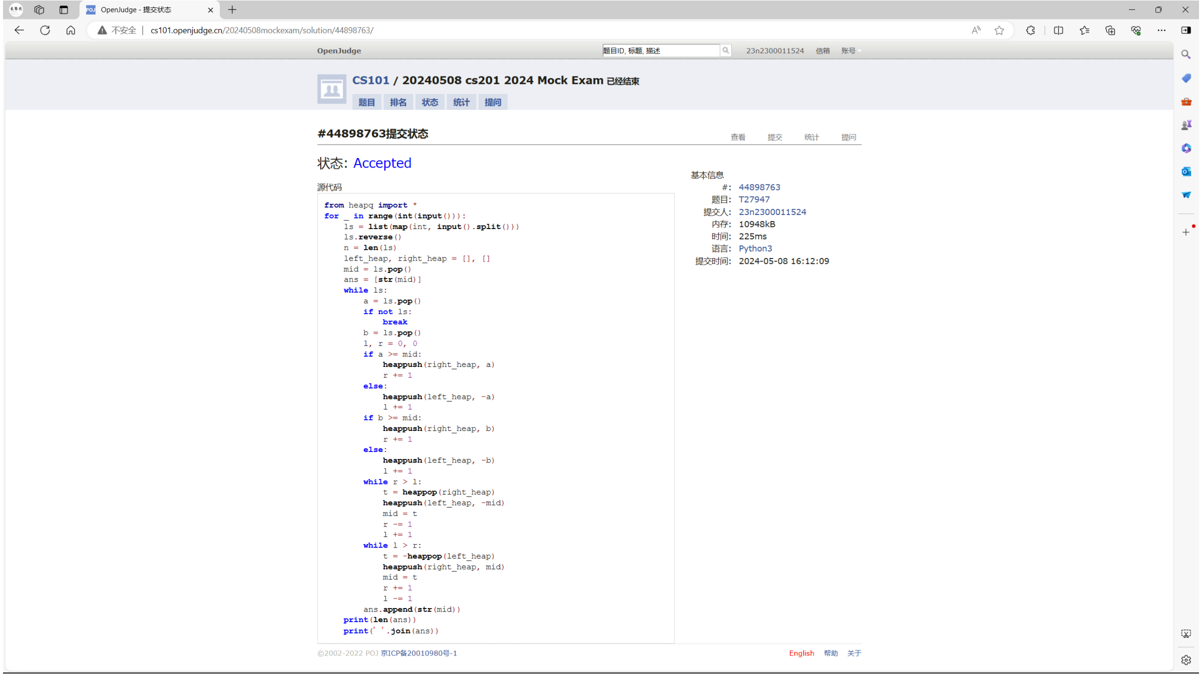
http://cs101.openjudge.cn/practice/27947/

思路：看着堆的提示想了半天才想起来左堆扔右堆，右堆仍左堆

代码

```
from heapq import *
for _ in range(int(input())):
    ls = list(map(int, input().split()))
    ls.reverse()
    n = len(ls)
    left_heap, right_heap = [], []
    mid = ls.pop()
    ans = [str(mid)]
    while ls:
        a = ls.pop()
        if not ls:
            break
        b = ls.pop()
        l, r = 0, 0
        if a >= mid:
            heappush(right_heap, a)
            r += 1
        else:
            heappush(left_heap, -a)
            l += 1
        if b >= mid:
            heappush(right_heap, b)
            r += 1
        else:
            heappush(left_heap, -b)
            l += 1
        while r > l:
            t = heappop(right_heap)
            heappush(left_heap, -mid)
            mid = t
            r -= 1
            l += 1
        while l > r:
            t = -heappop(left_heap)
            heappush(right_heap, mid)
            mid = t
            r += 1
            l -= 1
        ans.append(str(mid))
    print(len(ans))
    print(' '.join(ans))
```

代码运行截图



CS101 / 20240508 cs201 2024 Mock Exam 已经结束

#44898763提交状态

状态: Accepted

源代码

```
from heapq import *
for _ in range(int(input())):
    ls = list(map(int, input().split()))
    ls.reverse()
    n = len(ls)
    left_heap, right_heap = [], []
    mid = ls.pop()
    ans = [str(mid)]
    while ls:
        a = ls.pop()
        if not ls:
            break
        b = ls.pop()
        l, r = 0, 0
        if a >= mid:
            heappush(right_heap, a)
            r += 1
        else:
            heappush(left_heap, -a)
            l += 1
        if b >= mid:
            heappush(right_heap, b)
            r += 1
        else:
            heappush(left_heap, -b)
            l += 1
        while r > l:
            t = heappop(right_heap)
            heappush(left_heap, -mid)
            mid = t
            r -= 1
            l += 1
        while l > r:
            t = -heappop(left_heap)
            heappush(right_heap, mid)
            mid = t
            r += 1
            l -= 1
        ans.append(str(mid))
    print(len(ans))
    print(' '.join(ans))
```

基本信息

#: 44898763
题目: T27947
提交人: 23n2300011524
内存: 10948kB
时间: 225ms
语言: Python3
提交时间: 2024-05-08 16:12:09

©2002-2022 POJ 京ICP备20010980号-1

28190: 奶牛排队

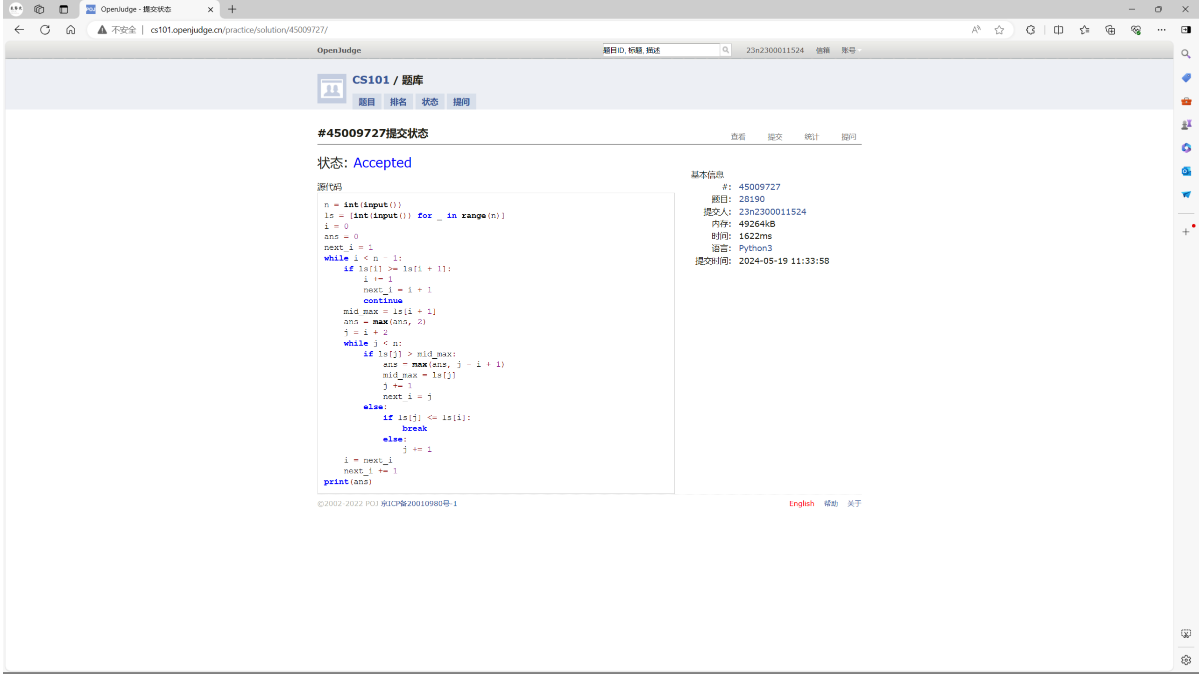
http://cs101.openjudge.cn/practice/28190/

思路：注意到某一处奶牛成为队尾后，队中任何一个奶牛如果作为队首不可能比当前队首的队列更长，如此可以加快速度

代码

```python
n = int(input())
ls = [int(input()) for _ in range(n)]
i = 0
ans = 0
next_i = 1
while i < n - 1:
    if ls[i] >= ls[i + 1]:
        i += 1
        next_i = i + 1
        continue
    mid_max = ls[i + 1]
    ans = max(ans, 2)
    j = i + 2
    while j < n:
        if ls[j] > mid_max:
            ans = max(ans, j - i + 1)
            mid_max = ls[j]
            j += 1
            next_i = j
        else:
            if ls[j] <= ls[i]:
                break
            else:
                j += 1
    i = next_i
    next_i += 1
print(ans)
```

代码运行截图



#45009727提交状态

状态: Accepted

源代码

```python
n = int(input())
ls = [int(input()) for _ in range(n)]
i = 0
ans = 0
next_i = 1
while i < n - 1:
    if ls[i] >= ls[i + 1]:
        i += 1
        next_i = i + 1
        continue
    mid_max = ls[i + 1]
    ans = max(ans, 2)
    j = i + 2
    while j < n:
        if ls[j] > mid_max:
            ans = max(ans, j - i + 1)
            mid_max = ls[j]
            j += 1
            next_i = j
        else:
            if ls[j] <= ls[i]:
                break
            else:
                j += 1
    i = next_i
    next_i += 1
print(ans)
```

基本信息

#:	45009727
题目:	28190
提交人:	23n2300011524
内存:	49264kB
时间:	1622ms
语言:	Python3
提交时间:	2024-05-19 11:33:58

©2002-2022 POJ 京ICP备20010980号-1

2. 学习总结和收获

因数据不够强侥幸通过奶牛排队，经历cupt和4个ddl大作战后开始补数算作业，一会补完笔试在学习下单调栈争取通过洛谷的测试数据。