Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Complied by ==田济维 物理学院

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

（python pycharm）

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

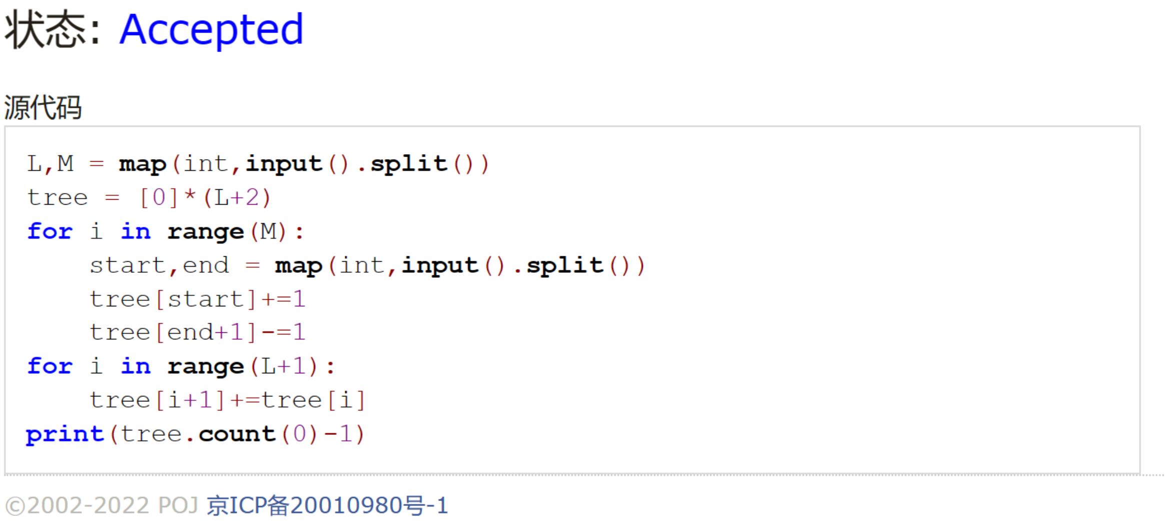
http://cs101.openjudge.cn/practice/02808/

思路：

代码

```python
#
L,M = map(int,input().split())
tree = [0]*(L+2)
for i in range(M):
    start,end = map(int,input().split())
    tree[start]+=1
    tree[end+1]-=1
for i in range(L+1):
    tree[i+1]+=tree[i]
print(tree.count(0)-1)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
L,M = map(int,input().split())
tree = [0]*(L+2)
for i in range(M):
    start,end = map(int,input().split())
    tree[start]+=1
    tree[end+1]-=1
for i in range(L+1):
    tree[i+1]+=tree[i]
print(tree.count(0)-1)
```

©2002-2022 POJ 京ICP备20010980号-1

20449: 是否被5整除

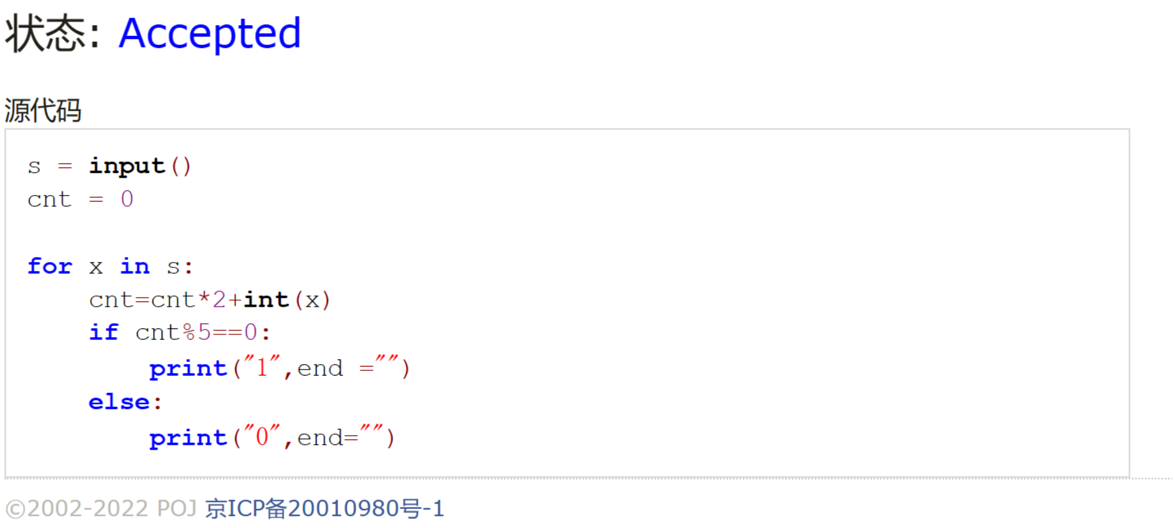
http://cs101.openjudge.cn/practice/20449/

思路：

代码

```python
# s = input()
cnt = 0

for x in s:
    cnt=cnt*2+int(x)
    if cnt%5==0:
        print("1",end ="")
    else:
        print("0",end="")

```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
s = input()
cnt = 0

for x in s:
    cnt=cnt*2+int(x)
    if cnt%5==0:
        print(”1”,end =””)
    else:
        print(”0”,end=””)
```

©2002-2022 POJ 京ICP备20010980号-1

01258: Agri-Net

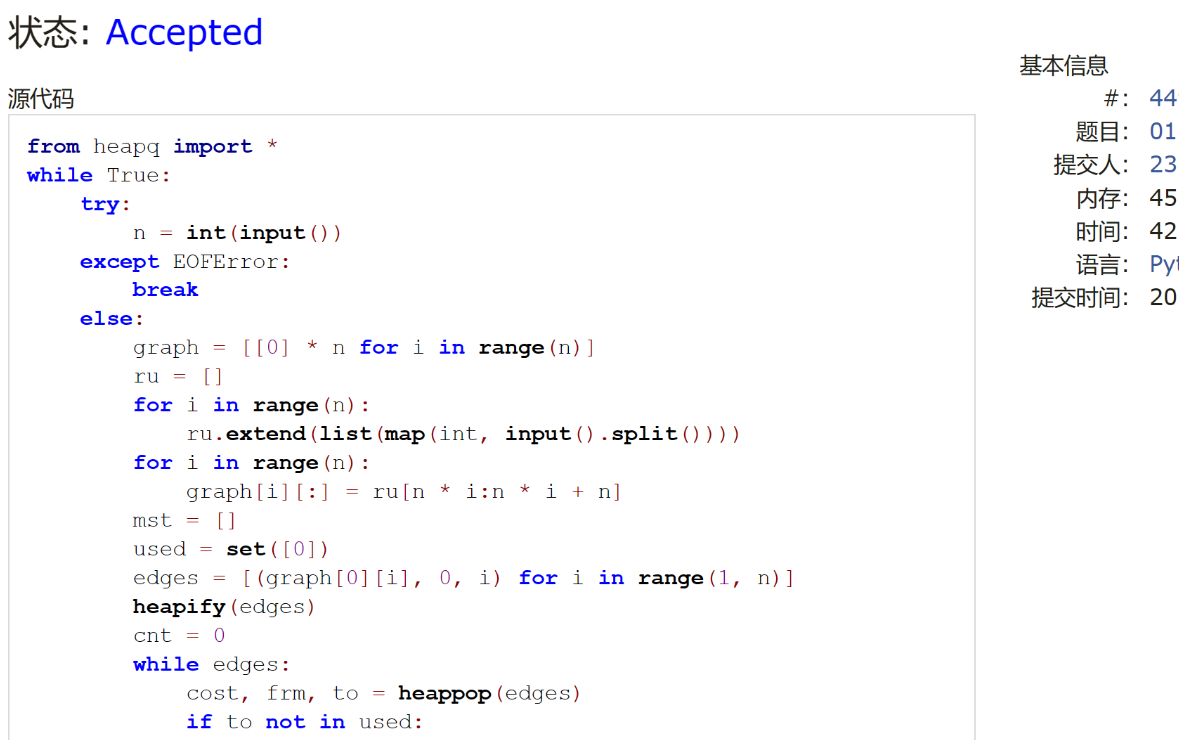
http://cs101.openjudge.cn/practice/01258/

思路：

代码

```
#
from heapq import *
while True:
    try:
        n = int(input())
    except EOFError:
        break
    else:
        graph = [[0] * n for i in range(n)]
        ru = []
        for i in range(n):
            ru.extend(list(map(int, input().split())))
        for i in range(n):
            graph[i][:] = ru[n * i:n * i + n]
        mst = []
        used = set([0])
        edges = [(graph[0][i], 0, i) for i in range(1, n)]
        heapify(edges)
        cnt = 0
        while edges:
            cost, frm, to = heappop(edges)
            if to not in used:
                used.add(to)
                cnt += cost
                for i in range(n):
                    if i != to and i not in used:
                        heappush(edges, (graph[to][i], to, i))
        print(cnt)

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
from heapq import *
while True:
    try:
        n = int(input())
    except EOFError:
        break
    else:
        graph = [[0] * n for i in range(n)]
        ru = []
        for i in range(n):
            ru.extend(list(map(int, input().split())))
        for i in range(n):
            graph[i][:] = ru[n * i:n * i + n]
        mst = []
        used = set([0])
        edges = [(graph[0][i], 0, i) for i in range(1, n)]
        heapify(edges)
        cnt = 0
        while edges:
            cost, frm, to = heappop(edges)
            if to not in used:
```

基本信息

#:	44
题目:	01
提交人:	23
内存:	45
时间:	42
语言:	Py
提交时间:	20

27635: 判断无向图是否连通有无回路(同23163)

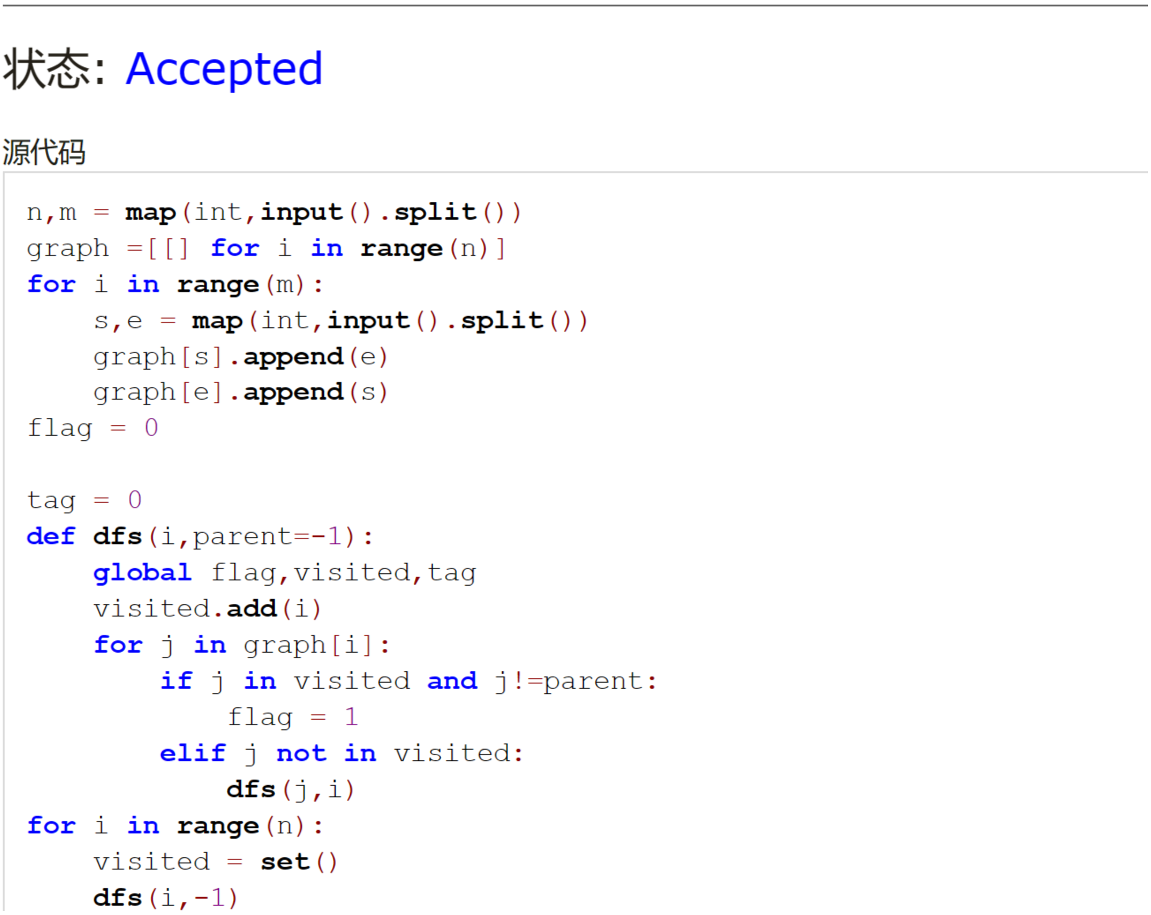
http://cs101.openjudge.cn/practice/27635/

思路：

代码

```
#
n,m = map(int,input().split())
graph =[[] for i in range(n)]
for i in range(m):
    s,e = map(int,input().split())
    graph[s].append(e)
    graph[e].append(s)
flag = 0

tag = 0
def dfs(i,parent=-1):
    global flag,visited,tag
    visited.add(i)
    for j in graph[i]:
        if j in visited and j!=parent:
            flag = 1
        elif j not in visited:
            dfs(j,i)
for i in range(n):
    visited = set()
    dfs(i,-1)
    if len(visited)==n:
        tag = 1
        if flag == 1:
            break

if tag:
    print("connected:yes")
else:
    print("connected:no")

if flag:
    print("loop:yes")
else:
    print("loop:no")
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
n,m = map(int,input().split())
graph =[[] for i in range(n)]
for i in range(m):
    s,e = map(int,input().split())
    graph[s].append(e)
    graph[e].append(s)
flag = 0

tag = 0
def dfs(i,parent=-1):
    global flag,visited,tag
    visited.add(i)
    for j in graph[i]:
        if j in visited and j!=parent:
            flag = 1
        elif j not in visited:
            dfs(j,i)
for i in range(n):
    visited = set()
    dfs(i,-1)
```

27947: 动态中位数

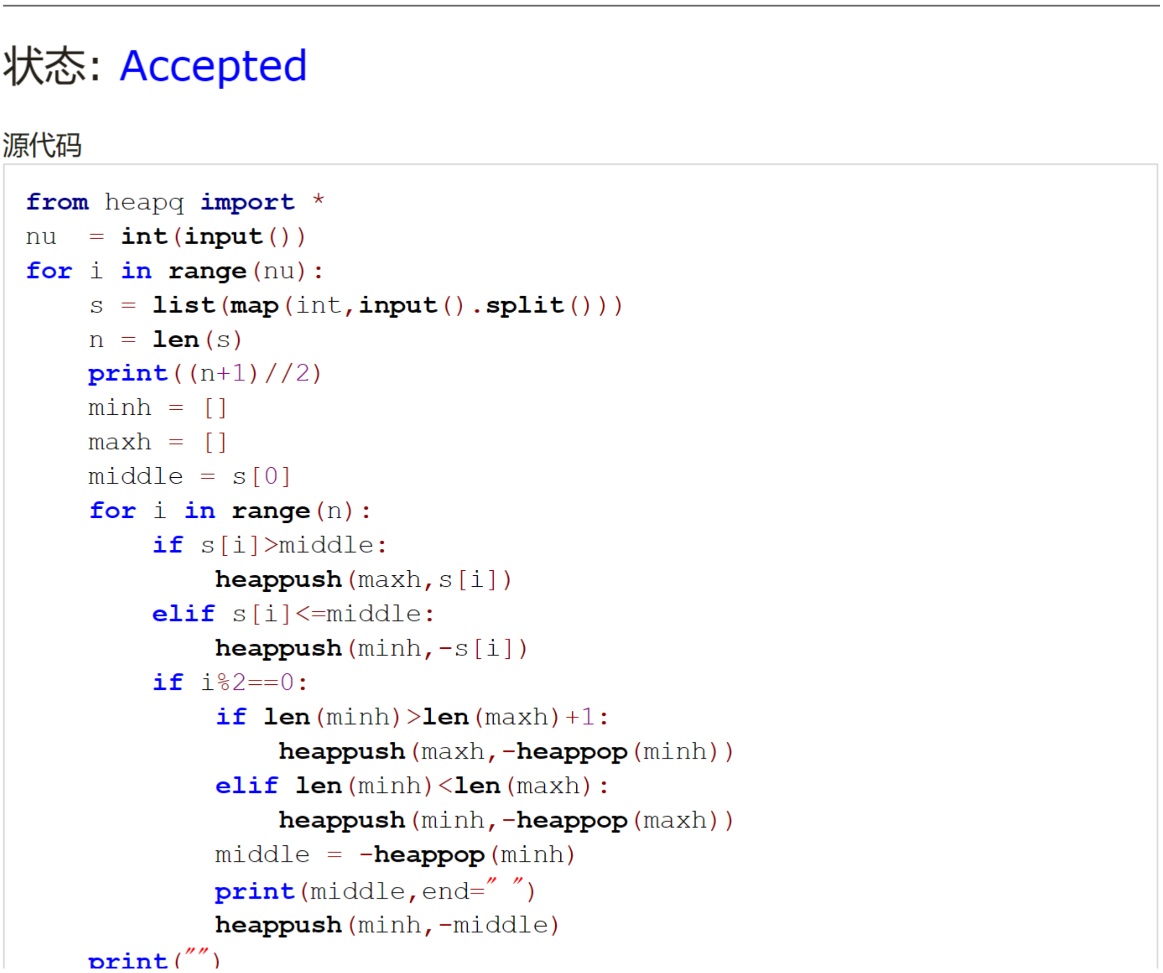
http://cs101.openjudge.cn/practice/27947/

思路：

代码

```python
#
from heapq import *
nu  = int(input())
for i in range(nu):
    s = list(map(int,input().split()))
    n = len(s)
    print((n+1)//2)
    minh = []
    maxh = []
    middle = s[0]
    for i in range(n):
        if s[i]>middle:
            heappush(maxh,s[i])
        elif s[i]<=middle:
            heappush(minh,-s[i])
        if i%2==0:
            if len(minh)>len(maxh)+1:
                heappush(maxh,-heappop(minh))
            elif len(minh)<len(maxh):
                heappush(minh,-heappop(maxh))
            middle = -heappop(minh)
            print(middle,end=" ")
            heappush(minh,-middle)
    print("")


```

代码运行截图（AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
from heapq import *
nu  = int(input())
for i in range(nu):
    s = list(map(int,input().split()))
    n = len(s)
    print((n+1)//2)
    minh = []
    maxh = []
    middle = s[0]
    for i in range(n):
        if s[i]>middle:
            heappush(maxh,s[i])
        elif s[i]<=middle:
            heappush(minh,-s[i])
        if i%2==0:
            if len(minh)>len(maxh)+1:
                heappush(maxh,-heappop(minh))
            elif len(minh)<len(maxh):
                heappush(minh,-heappop(maxh))
            middle = -heappop(minh)
            print(middle,end=" ")
            heappush(minh,-middle)
    print("")
```

28190: 奶牛排队

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

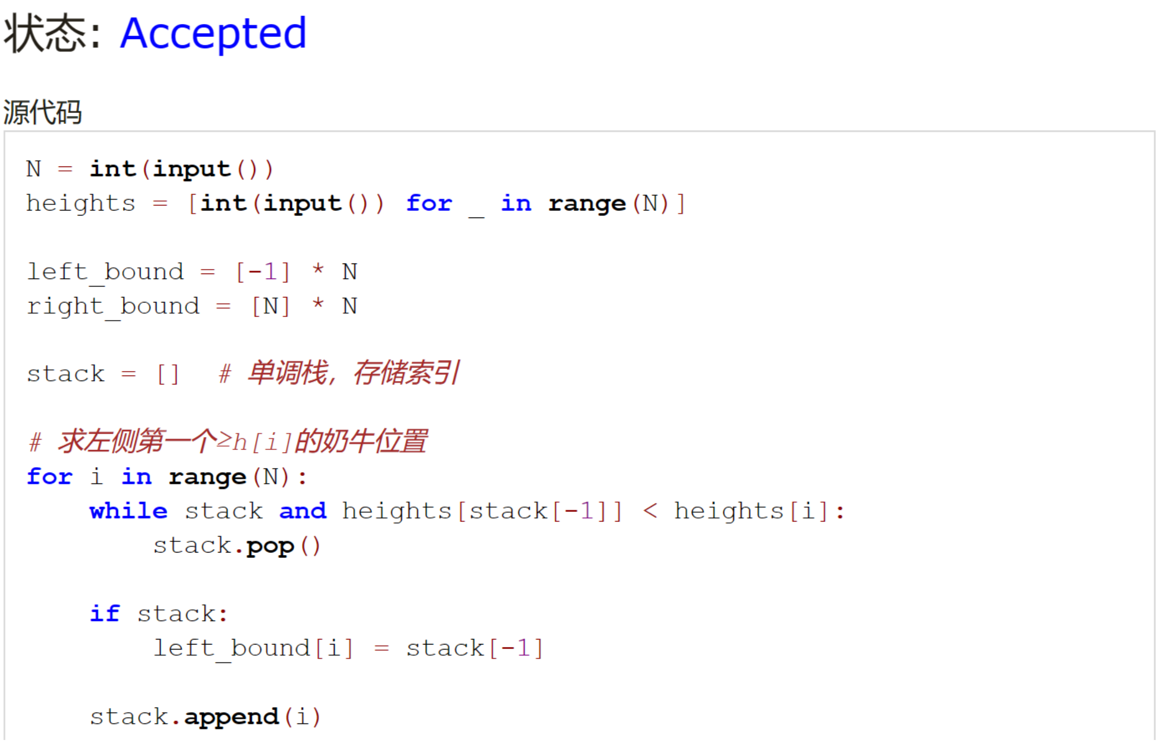
思路：不会做，学习了单调栈

代码

```
#
N = int(input())
heights = [int(input()) for _ in range(N)]

left_bound = [-1] * N
right_bound = [N] * N

stack = []  # 单调栈，存储索引

# 求左侧第一个≥h[i]的奶牛位置
for i in range(N):
    while stack and heights[stack[-1]] < heights[i]:
        stack.pop()

    if stack:
        left_bound[i] = stack[-1]
```

```python

    stack.append(i)

stack = []  # 清空栈以供寻找右边界使用

# 求右侧第一个≤h[i]的奶牛位
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-1, -1, -1):  # 从大到小枚举是个技巧
#     for j in range(left_bound[i] + 1, i):
#         if right_bound[j] > i:
#             ans = max(ans, i - j + 1)
#             break
#
#     if i <= ans:
#         break

for i in range(N):  # 枚举右端点 B寻找 A，更新 ans
    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"）



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 = []   # 单调栈，存储索引

# 求左侧第一个≥h[i]的奶牛位置
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)
```

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。

学习了单调栈的性质