Day-49

#!/bin/python3

import math

import os

import random

import re

import sys

#

# Complete the 'climbingLeaderboard' function below.

#

# The function is expected to return an INTEGER\_ARRAY.

# The function accepts following parameters:

#  1. INTEGER\_ARRAY ranked

#  2. INTEGER\_ARRAY player

#

def climbingLeaderboard(ranked, player):

    # Write your code here

    player.sort(reverse = True)

    res = []

    index = 0

    ranked = list(dict.fromkeys(ranked))

    for i in player:

        while index < len(ranked) and ranked[index] > i:

            index += 1

        if index >= len(ranked):

            index = len(ranked)

            res.append(index + 1)

            continue

        res.append(index + 1)

    res.reverse()

    return res

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    ranked\_count = int(input().strip())

    ranked = list(map(int, input().rstrip().split()))

    player\_count = int(input().strip())

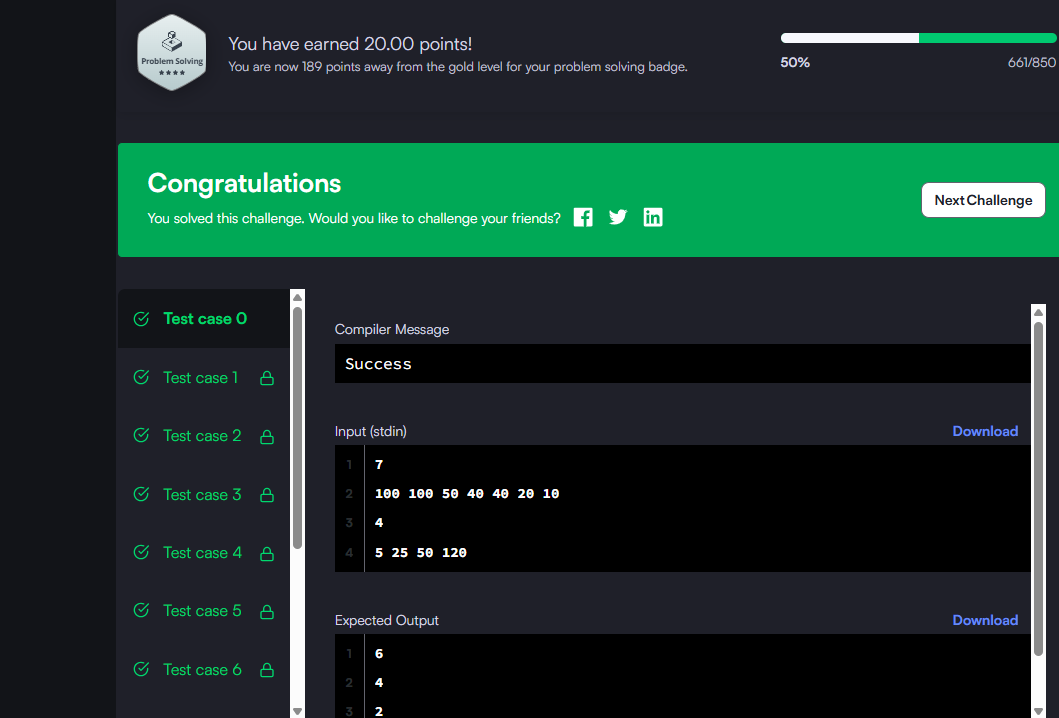
    player = list(map(int, input().rstrip().split()))

    result = climbingLeaderboard(ranked, player)

    fptr.write('\n'.join(map(str, result)))

    fptr.write('\n')

    fptr.close()



import heapq

from typing import List

class Solution:

def maxEvents(self, events: List[List[int]]) -> int:

# Step 1: sort events by start day

events.sort()

# Min-heap for event end days

heap = []

res = 0

i = 0

n = len(events)

# Day loop: from 1 to max end day

day = 1

while i < n or heap:

# If heap is empty, jump to next event's start

if not heap and i < n:

day = events[i][0]

# Add all events starting today

while i < n and events[i][0] == day:

heapq.heappush(heap, events[i][1])

i += 1

# Remove expired events

while heap and heap[0] < day:

heapq.heappop(heap)

# Attend one event (earliest ending one)

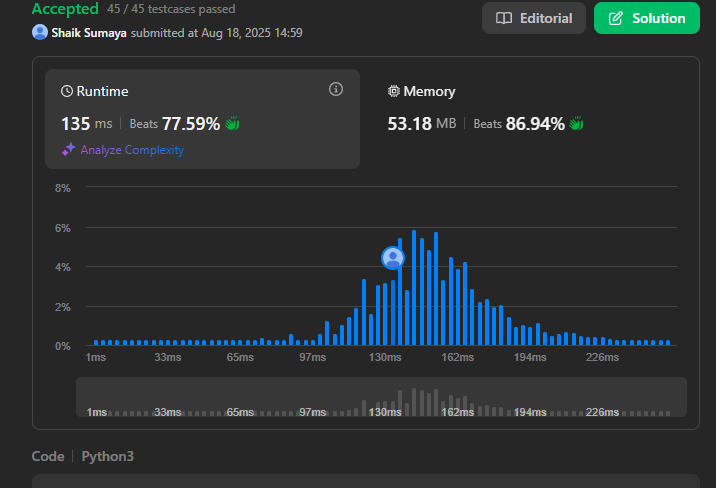
if heap:

heapq.heappop(heap)

res += 1

day += 1

return res



import sys

data = sys.stdin.read().strip().split()

t = int(data[0])

idx = 1

for \_ in range(t):

n = int(data[idx])

idx += 1

balls = list(map(int, data[idx:idx+n]))

idx += n

print(max(balls))

