Climbing the Leaderboard

64.8 more points to get your gold badge!

Rank: 281538 | Points: 785.2/850

Your Climbing the Leaderboard submission got 20.00 points. Share Post

You are now 64.8 points away from the gold level for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

An arcade game player wants to climb to the top of the leaderboard and track their ranking. The game uses Dense Ranking, so its leaderboard works like this:

- The player with the highest score is ranked number 1 on the leaderboard.
- Players who have equal scores receive the same ranking number, and the next player(s) receive the immediately following ranking number.

Example

$ranked = [100, 90, 90, 80]$

$player = [70, 80, 105]$

The ranked players will have ranks 1, 2, 2, and 3, respectively. If the player's scores are 70, 80 and 105, their rankings after each game are 4^{th}, 3^{rd} and 1^{st}. Return $[4, 3, 1]$.

Function Description

Complete the climbingLeaderboard function in the editor below.

climbingLeaderboard has the following parameter(s):

- int ranked[n]: the leaderboard scores
- int player[m]: the player's scores

Returns

- int[m]: the player's rank after each new score

Input Format

The first line contains an integer n, the number of players on the leaderboard.

The next line contains n space-separated integers $ranked[i]$, the leaderboard scores in decreasing order.

The next line contains an integer, m, the number games the player plays.

The last line contains m space-separated integers $player[j]$, the game scores.

Constraints

- $1 \le n \le 2 \times 10^5$
- $1 \le m \le 2 \times 10^5$
- $0 \le ranked[i] \le 10^9$ for $0 \le i < n$
- $0 \le player[j] \le 10^9$ for $0 \le j < m$
- The existing leaderboard, $ranked$, is in descending order.
- The player's scores, $player$, are in ascending order.

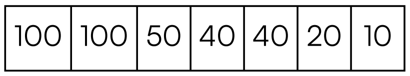
Subtask

For 60% of the maximum score:

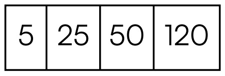
- $1 \le n \le 200$
- $1 \le m \le 200$

Sample Input 1 Copy Download

```
7
100 100 50 40 40 20 10
4
5 25 50 120
```



Array: ranked



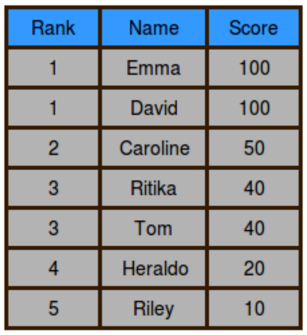
Array: player

Sample Output 1

```
6
4
2
1
```

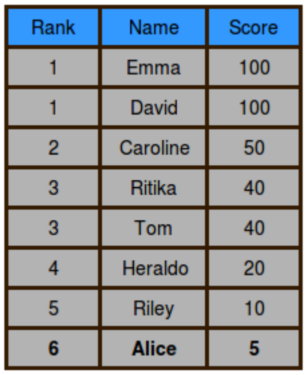
Explanation 1

Alice starts playing with 7 players already on the leaderboard, which looks like this:



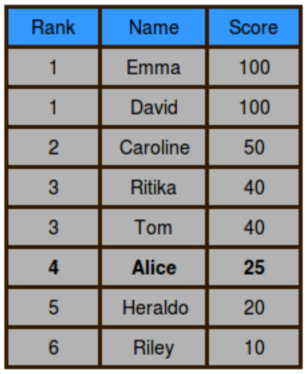
Rank	Name	Score
1	Emma	100
1	David	100
2	Caroline	50
3	Ritika	40
3	Tom	40
4	Heraldo	20
5	Riley	10

After Alice finishes game 0, her score is 5 and her ranking is 6:



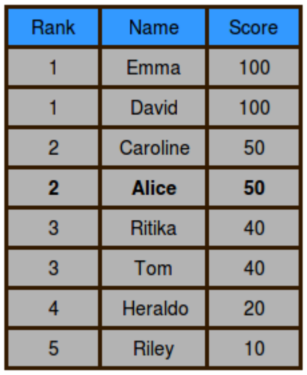
Rank	Name	Score
1	Emma	100
1	David	100
2	Caroline	50
3	Ritika	40
3	Tom	40
4	Heraldo	20
5	Riley	10
6	**Alice**	**5**

After Alice finishes game 1, her score is 25 and her ranking is 4:



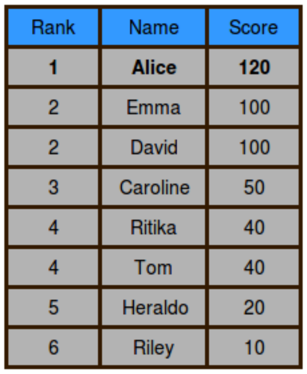
Rank	Name	Score
1	Emma	100
1	David	100
2	Caroline	50
3	Ritika	40
3	Tom	40
4	**Alice**	**25**
5	Heraldo	20
6	Riley	10

After Alice finishes game 2, her score is 50 and her ranking is tied with Caroline at 2:



Rank	Name	Score
1	Emma	100
1	David	100
2	Caroline	50
2	**Alice**	**50**
3	Ritika	40
3	Tom	40
4	Heraldo	20
5	Riley	10

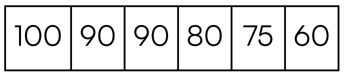
After Alice finishes game 3, her score is 120 and her ranking is 1:



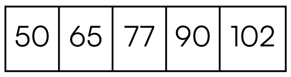
Rank	Name	Score
1	**Alice**	**120**
2	Emma	100
2	David	100
3	Caroline	50
4	Ritika	40
4	Tom	40
5	Heraldo	20
6	Riley	10

Sample Input 2

Copy Download



Array: ranked



Array: player

```
6
100 90 90 80 75 60
5
50 65 77 90 102
```

Sample Output 2

```
6
5
4
2
1
```

Change Theme Language Python 3

```python
import re
import sys
import bisect

#
# 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
    unique_sorted = sorted(list(set(ranked)), reverse=True)

    result = []
    for score in player:
        left, right = 0, len(unique_sorted)
        # Binary search to find the insertion point
        while left < right:
            mid = (left + right) // 2
            if unique_sorted[mid] > score:
                left = mid + 1
            else:
                right = mid
        result.append(left + 1)
    return result


if   name   == '  main  ':
```

EMACS Line: 38 Col: 5

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!

You are now 64.8 points away from the gold level for your problem solving badge.

83% 785.2/850

Problem Solving
★★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- ✓ **Test case 0**
- ✓ Test case 1
- ✓ Test case 2
- ✓ Test case 3
- ✓ Test case 4
- ✓ Test case 5
- ✓ Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

```
7
100 100 50 40 40 20 10
4
5 25 50 120
```

Expected Output Download

```
6
4
2
```

Blog | Scoring | Environment | FAQ | About Us | Helpdesk | Careers | Terms Of Service | Privacy Policy