

## 2410. Maximum Matching of Players With Trainers

[My Submissions \(/contest/biweekly-contest-87/problems/maximum-matching-of-players-with-trainers/submissions/\)](#)
[Back to Contest \(/contest/biweekly-contest-87/\)](#)

You are given a **0-indexed** integer array `players`, where `players[i]` represents the **ability** of the  $i^{\text{th}}$  player. You are also given a **0-indexed** integer array `trainers`, where `trainers[j]` represents the **training capacity** of the  $j^{\text{th}}$  trainer.

The  $i^{\text{th}}$  player can **match** with the  $j^{\text{th}}$  trainer if the player's ability is **less than or equal to** the trainer's training capacity. Additionally, the  $i^{\text{th}}$  player can be matched with at most one trainer, and the  $j^{\text{th}}$  trainer can be matched with at most one player.

Return the **maximum** number of matchings between players and trainers that satisfy these conditions.

User Accepted:	9867
User Tried:	11124
Total Accepted:	10182
Total Submissions:	18846
Difficulty:	Medium

### Example 1:

**Input:** `players = [4,7,9]`, `trainers = [8,2,5,8]`

**Output:** 2

**Explanation:**

One of the ways we can form two matchings is as follows:

- `players[0]` can be matched with `trainers[0]` since  $4 \leq 8$ .
- `players[1]` can be matched with `trainers[3]` since  $7 \leq 8$ .

It can be proven that 2 is the maximum number of matchings that can be formed.

### Example 2:

**Input:** `players = [1,1,1]`, `trainers = [10]`

**Output:** 1

**Explanation:**

The trainer can be matched with any of the 3 players.

Each player can only be matched with one trainer, so the maximum answer is 1.

### Constraints:

- $1 \leq \text{players.length}, \text{trainers.length} \leq 10^5$
- $1 \leq \text{players}[i], \text{trainers}[j] \leq 10^9$

[Discuss \(https://leetcode.com/problems/maximum-matching-of-players-with-trainers/discuss/\)](https://leetcode.com/problems/maximum-matching-of-players-with-trainers/discuss/)

Java



```

1 class Solution {
2     public int matchPlayersAndTrainers(int[] a, int[] b) {
3         TreeMap<Integer, Integer> m = counter(b);
4         int res = 0;
5         for (int x : a) {
6             Integer next = m.ceilingKey(x);
7             if (next != null) {
8                 res++;
9                 removeOneOrManyMap(m, next);
10            }
11        }
12        return res;
13    }
14
15    TreeMap<Integer, Integer> counter(int[] a) {
16        TreeMap<Integer, Integer> m = new TreeMap<>();
17        for (int x : a) m.put(x, m.getOrDefault(x, 0) + 1);
18        return m;
19    }
20
21    <T> void removeOneOrManyMap(TreeMap<T, Integer> m, T x, int... args) {
22        int cnt = args.length == 0 ? 1 : args[0], occ = m.get(x);
23        if (occ > cnt) {

```

```
24         m.put(x, occ - cnt);  
25     } else {  
26         m.remove(x);  
27     }  
28 }  
29 }
```

☐ Custom Testcase[Use Example Testcases](#)[▶ Run](#)[🚀 Submit](#)**Submission Result: Accepted** (</submissions/detail/802366992/>) ⓘ[More Details ▶ \(/submissions/detail/802366992/\)](/submissions/detail/802366992/)

Share your acceptance!

Copyright © 2022 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs/\)](/jobs/) | [Bug Bounty \(/bugbounty/\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student/\)](/student/) | [Terms \(/terms/\)](/terms/) | [Privacy Policy \(/privacy/\)](/privacy/) [United States \(/region/\)](/region/)