Problem 2070: Most Beautiful Item for Each Query

Problem Information

Difficulty: Medium
Acceptance Rate: 62.07%
Paid Only: No
Tags: Array, Binary Search, Sorting

Problem Description

You are given a 2D integer array `items` where `items[i] = [pricei, beautyi]` denotes the **price** and **beauty** of an item respectively.

You are also given a **0-indexed** integer array `queries`. For each `queries[j]`, you want to determine the **maximum beauty** of an item whose **price** is **less than or equal** to `queries[j]`. If no such item exists, then the answer to this query is `0`.

Return _an array_`answer` _of the same length as_`queries` _where_`answer[j]`_is the answer to the_`jth` _query_.

Example 1:

Input: items = [[1,2],[3,2],[2,4],[5,6],[3,5]], queries = [1,2,3,4,5,6] **Output:** [2,4,5,5,6,6] **Explanation:** - For queries[0]=1, [1,2] is the only item which has price <= 1. Hence, the answer for this query is 2. - For queries[1]=2, the items which can be considered are [1,2] and [2,4]. The maximum beauty among them is 4. - For queries[2]=3 and queries[3]=4, the items which can be considered are [1,2], [3,2], [2,4], and [3,5]. The maximum beauty among them is 5. - For queries[4]=5 and queries[5]=6, all items can be considered. Hence, the answer for them is the maximum beauty of all items, i.e., 6.

Example 2:

Input: items = [[1,2],[1,2],[1,3],[1,4]], queries = [1] **Output:** [4] **Explanation:** The price of every item is equal to 1, so we choose the item with the maximum beauty 4. Note that multiple items can have the same price and/or beauty.

Example 3:

Input: items = [[10,1000]], queries = [5] **Output:** [0] **Explanation:** No item has a price less than or equal to 5, so no item can be chosen. Hence, the answer to the query is 0.

Constraints:

* `1 <= items.length, queries.length <= 105` * `items[i].length == 2` * `1 <= pricei, beautyi, queries[j] <= 109`

Code Snippets

C++:

```
class Solution {
public:
vector<int> maximumBeauty(vector<vector<int>>& items, vector<int>& queries) {

}
};
```

Java:

```
class Solution {
public int[] maximumBeauty(int[][] items, int[] queries) {

}
}
```

Python3:

```
class Solution:
def maximumBeauty(self, items: List[List[int]], queries: List[int]) ->
List[int]:
```