Problem 2819: Minimum Relative Loss After Buying Chocolates

Problem Information

Difficulty: Hard
Acceptance Rate: 45.44%
Paid Only: Yes
Tags: Array, Binary Search, Sorting, Prefix Sum

Problem Description

You are given an integer array `prices`, which shows the chocolate prices and a 2D integer array `queries`, where `queries[i] = [ki, mi]`.

Alice and Bob went to buy some chocolates, and Alice suggested a way to pay for them, and Bob agreed.

The terms for each query are as follows:

* If the price of a chocolate is **less than or equal to** `ki`, Bob pays for it. * Otherwise, Bob pays `ki` of it, and Alice pays the **rest**.

Bob wants to select **exactly** `mi` chocolates such that his **relative loss** is **minimized** , more formally, if, in total, Alice has paid `ai` and Bob has paid `bi`, Bob wants to minimize `bi - ai`.

Return _an integer array_ `ans` _where_ `ans[i]` _is Bob 's **minimum relative loss** possible for_ `queries[i]`.

Example 1:

Input: prices = [1,9,22,10,19], queries = [[18,4],[5,2]] **Output:** [34,-21] **Explanation:** For the 1st query Bob selects the chocolates with prices [1,9,10,22]. He pays 1 + 9 + 10 + 18 = 38 and Alice pays 0 + 0 + 0 + 4 = 4. So Bob's relative loss is 38 - 4 = 34. For the 2nd query Bob selects the chocolates with prices [19,22]. He pays 5 + 5 = 10 and Alice pays 14 + 17 = 31. So Bob's relative loss is 10 - 31 = -21. It can be shown that these are the minimum

possible relative losses.

Example 2:

Input: prices = [1,5,4,3,7,11,9], queries = [[5,4],[5,7],[7,3],[4,5]] **Output:** [4,16,7,1] **Explanation:** For the 1st query Bob selects the chocolates with prices [1,3,9,11]. He pays 1 + 3 + 5 + 5 = 14 and Alice pays 0 + 0 + 4 + 6 = 10. So Bob's relative loss is 14 - 10 = 4. For the 2nd query Bob has to select all the chocolates. He pays 1 + 5 + 4 + 3 + 5 + 5 + 5 = 28 and Alice pays 0 + 0 + 0 + 0 + 2 + 6 + 4 = 12. So Bob's relative loss is 28 - 12 = 16. For the 3rd query Bob selects the chocolates with prices [1,3,11] and he pays 1 + 3 + 7 = 11 and Alice pays 0 + 0 + 4 = 4. So Bob's relative loss is 11 - 4 = 7. For the 4th query Bob selects the chocolates with prices [1,3,7,9,11] and he pays 1 + 3 + 4 + 4 + 4 = 16 and Alice pays 0 + 0 + 3 + 5 + 7 = 15. So Bob's relative loss is 16 - 15 = 1. It can be shown that these are the minimum possible relative losses.

Example 3:

Input: prices = [5,6,7], queries = [[10,1],[5,3],[3,3]] **Output:** [5,12,0] **Explanation:** For the 1st query Bob selects the chocolate with price 5 and he pays 5 and Alice pays 0. So Bob's relative loss is 5 - 0 = 5. For the 2nd query Bob has to select all the chocolates. He pays 5 + 5 + 5 = 15 and Alice pays 0 + 1 + 2 = 3. So Bob's relative loss is 15 - 3 = 12. For the 3rd query Bob has to select all the chocolates. He pays 3 + 3 + 3 = 9 and Alice pays 2 + 3 + 4 = 9. So Bob's relative loss is 9 - 9 = 0. It can be shown that these are the minimum possible relative losses.

Constraints:

* `1 <= prices.length == n <= 105` * `1 <= prices[i] <= 109` * `1 <= queries.length <= 105` * `queries[i].length == 2` * `1 <= ki <= 109` * `1 <= mi <= n`

Code Snippets

C++:

```
class Solution {
public:
vector<long long> minimumRelativeLosses(vector<int>& prices,
vector<vector<int>>& queries) {

}
```

```
};
```

Java:

```
class Solution {
public long[] minimumRelativeLosses(int[] prices, int[][] queries) {

}
}
```

Python3:

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