Description | Editorial | Solutions | Submissions

1086. High Five

Premium

Solved

Easy | Topics | Companies | Hint

Given a list of the scores of different students, `items`, where `items[i] = [ID_i, score_i]` represents one score from a student with `ID_i`, calculate each student's **top five average**.

Return *the answer as an array of pairs* `result`, *where* `result[j] = [ID_j, topFiveAverage_j]` *represents the student with* `ID_j` *and their* ***top five average****. Sort* `result` *by* `ID_j` *in* ***increasing order****.*

A student's **top five average** is calculated by taking the sum of their top five scores and dividing it by `5` using **integer division**.

Example 1:

```
Input: items = [[1,91],[1,92],[2,93],[2,97],[1,60],[2,77],[1,65],[1,87],[1,100],[2,100],[2,76]]
Output: [[1,87],[2,88]]
Explanation:
The student with ID = 1 got scores 91, 92, 60, 65, 87, and 100. Their top five average is (100 + 92 + 91 + 87 + 65) / 5 = 87.
The student with ID = 2 got scores 93, 97, 77, 100, and 76. Their top five average is (100 + 97 + 93 + 77 + 76) / 5 = 88.6, but with integer division their average converts to 88.
```

Example 2:

```
Input: items = [[1,100],[7,100],[1,100],[7,100],[1,100],[7,100],[1,100],[7,100],[1,100],[7,100]]
Output: [[1,100],[7,100]]
```

Constraints:

- `1 <= items.length <= 1000`
- `items[i].length == 2`
- `1 <= ID_i <= 1000`
- `0 <= score_i <= 100`
- For each `ID_i`, there will be **at least** five scores.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 101.6K | Submissions 136.3K | Acceptance Rate 74.6%

Topics

Companies

Hint 1

Hint 2

Hint 3

Similar Questions

Discussion (4)

788 | 4

Code

Python3 • Auto

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

        dictx = {}

        for each in items:
            if each[0] not in dictx:
                dictx[each[0]] = [each[1]]
            else:
                dictx[each[0]].append(each[1])

        print(dictx)

        ans = []

        for key, value in dictx.items():
            scores = sorted(value, reverse=True)[:5]
```

Saved

Ln 14, Col 17

Testcase | Test Result

Input

items =

```
[[1,91],[1,92],[2,93],[2,97],[1,60],[2,77],[1,65],[1,87],[1,100],[2,100],[2,76]]
```

Stdout

```
{1: [91, 92, 60, 65, 87, 100], 2: [93, 97, 77, 100, 76]}
```

Output

```
[[1,87],[2,88]]
```

Expected

```
[[1,87],[2,88]]
```

Contribute a testcase