Problem 2284: Sender With Largest Word Count

Problem Information

Difficulty: Medium
Acceptance Rate: 59.12%
Paid Only: No
Tags: Array, Hash Table, String, Counting

Problem Description

You have a chat log of `n` messages. You are given two string arrays `messages` and `senders` where `messages[i]` is a **message** sent by `senders[i]`.

A **message** is list of **words** that are separated by a single space with no leading or trailing spaces. The **word count** of a sender is the total number of **words** sent by the sender. Note that a sender may send more than one message.

Return _the sender with the**largest** word count_. If there is more than one sender with the largest word count, return _the one with the**lexicographically largest** name_.

Note:

* Uppercase letters come before lowercase letters in lexicographical order. * `"Alice"` and `"alice"` are distinct.

Example 1:

Input: messages = ["Hello userTwooo","Hi userThree","Wonderful day Alice","Nice day userThree"], senders = ["Alice","userTwo","userThree","Alice"] **Output:** "Alice" **Explanation:** Alice sends a total of 2 + 3 = 5 words. userTwo sends a total of 2 words. userThree sends a total of 3 words. Since Alice has the largest word count, we return "Alice".

Example 2:

Input: messages = ["How is leetcode for everyone","Leetcode is useful for practice"], senders = ["Bob","Charlie"] **Output:** "Charlie" **Explanation:** Bob sends a total of 5

words. Charlie sends a total of 5 words. Since there is a tie for the largest word count, we return the sender with the lexicographically larger name, Charlie.

Constraints:

* `n == messages.length == senders.length` * `1 <= n <= 104` * `1 <= messages[i].length <= 100` * `1 <= senders[i].length <= 10` * `messages[i]` consists of uppercase and lowercase English letters and `' '`. * All the words in `messages[i]` are separated by **a single space**. * `messages[i]` does not have leading or trailing spaces. * `senders[i]` consists of uppercase and lowercase English letters only.

Code Snippets

C++:

```
class Solution {
public:
string largestWordCount(vector<string>& messages, vector<string>& senders) {

}
};
```

Java:

```
class Solution {
public String largestWordCount(String[] messages, String[] senders) {

}
}
```

Python3:

```
class Solution:
def largestWordCount(self, messages: List[str], senders: List[str]) -> str:
```