Problem 3606: Coupon Code Validator

Problem Information

Difficulty: Easy
Acceptance Rate: 53.74%
Paid Only: No
Tags: Array, Hash Table, String, Sorting

Problem Description

You are given three arrays of length `n` that describe the properties of `n` coupons: `code`, `businessLine`, and `isActive`. The `ith `coupon has:

* `code[i]`: a **string** representing the coupon identifier. * `businessLine[i]`: a **string** denoting the business category of the coupon. * `isActive[i]`: a **boolean** indicating whether the coupon is currently active.

A coupon is considered **valid** if all of the following conditions hold:

1. `code[i]` is non-empty and consists only of alphanumeric characters (a-z, A-Z, 0-9) and underscores (`_`). 2. `businessLine[i]` is one of the following four categories: `"electronics"`, `"grocery"`, `"pharmacy"`, `"restaurant"`. 3. `isActive[i]` is **true**.

Return an array of the **codes** of all valid coupons, **sorted** first by their **businessLine** in the order: `"electronics"`, `"grocery"`, `"pharmacy", "restaurant"`, and then by **code** in lexicographical (ascending) order within each category.

Example 1:

Input: code = ["SAVE20","","PHARMA5","SAVE@20"], businessLine = ["restaurant","grocery","pharmacy","restaurant"], isActive = [true,true,true,true]

Output: ["PHARMA5","SAVE20"]

Explanation:

* First coupon is valid. * Second coupon has empty code (invalid). * Third coupon is valid. * Fourth coupon has special character `@` (invalid).

Example 2:

Input: code = ["GROCERY15","ELECTRONICS_50","DISCOUNT10"], businessLine = ["grocery","electronics","invalid"], isActive = [false,true,true]

Output: ["ELECTRONICS_50"]

Explanation:

* First coupon is inactive (invalid). * Second coupon is valid. * Third coupon has invalid business line (invalid).

Constraints:

* `n == code.length == businessLine.length == isActive.length` * `1 <= n <= 100` * `0 <= code[i].length, businessLine[i].length <= 100` * `code[i]` and `businessLine[i]` consist of printable ASCII characters. * `isActive[i]` is either `true` or `false`.

Code Snippets

C++:

```
class Solution {
public:
vector<string> validateCoupons(vector<string>& code, vector<string>&
businessLine, vector<bool>& isActive) {

}
};
```

Java:

```
class Solution {
public List<String> validateCoupons(String[] code, String[] businessLine,
boolean[] isActive) {

}
```

```
}
```

Python3:

```
class Solution:
def validateCoupons(self, code: List[str], businessLine: List[str], isActive:
List[bool]) -> List[str]:
```