Problem 3078: Match Alphanumerical Pattern in Matrix I

Problem Information

Difficulty: Medium
Acceptance Rate: 64.41%
Paid Only: Yes
Tags: Array, Hash Table, String, Matrix

Problem Description

You are given a 2D integer matrix `board` and a 2D character matrix `pattern`. Where `0 <= board[r][c] <= 9` and each element of `pattern` is either a digit or a lowercase English letter.

Your task is to find a submatrix of `board` that **matches** `pattern`.

An integer matrix `part` matches `pattern` if we can replace cells containing letters in `pattern` with some digits (each **distinct** letter with a **unique** digit) in such a way that the resulting matrix becomes identical to the integer matrix `part`. In other words,

* The matrices have identical dimensions. * If `pattern[r][c]` is a digit, then `part[r][c]` must be the **same** digit. * If `pattern[r][c]` is a letter `x`: * For every `pattern[i][j] == x`, `part[i][j]` must be the **same** as `part[r][c]`. * For every `pattern[i][j] != x`, `part[i][j]` must be **different** than `part[r][c]`.

Return _an array of length_`2` _containing the row number and column number of the upper-left corner of a submatrix of_`board` _which matches_`pattern` _. If there is more than one such submatrix, return the coordinates of the submatrix with the lowest row index, and in case there is still a tie, return the coordinates of the submatrix with the lowest column index. If there are no suitable answers, return_ `[-1, -1]`.

Example 1:

1 | 2 | 2 ---|---|--- 2 | 2 | 3 2 | 3 | 3 a | b ---|--- b | b **Input:** board = [[1,2,2],[2,2,3],[2,3,3]], pattern = ["ab","bb"]

Output: [0,0]

Explanation: If we consider this mapping: `"a" -> 1` and `"b" -> 2`; the submatrix with the upper-left corner `(0,0)` is a match as outlined in the matrix above.

Note that the submatrix with the upper-left corner (1,1) is also a match but since it comes after the other one, we return `[0,0]`.

Example 2:

1 | 1 | 2 ---|---|--- 3 | 3 | 4 6 | 6 | 6 a | b ---|--- 6 | 6 **Input:** board = [[1,1,2],[3,3,4],[6,6,6]], pattern = ["ab","66"]

Output: [1,1]

Explanation: If we consider this mapping: `"a" -> 3` and `"b" -> 4`; the submatrix with the upper-left corner `(1,1)` is a match as outlined in the matrix above.

Note that since the corresponding values of `"a"` and `"b"` must differ, the submatrix with the upper-left corner `(1,0)` is not a match. Hence, we return `[1,1]`.

Example 3:

1 | 2 ---|--- 2 | 1 x | x ---|--- **Input:** board = [[1,2],[2,1]], pattern = ["xx"]

Output: [-1,-1]

Explanation: Since the values of the matched submatrix must be the same, there is no match. Hence, we return `[-1,-1]`.

Constraints:

* `1 <= board.length <= 50` * `1 <= board[i].length <= 50` * `0 <= board[i][j] <= 9` * `1 <= pattern.length <= 50` * `1 <= pattern[i].length <= 50` * `pattern[i][j]` is either a digit represented as a string or a lowercase English letter.

Code Snippets

C++:

```
class Solution {
public:
vector<int> findPattern(vector<vector<int>>& board, vector<string>& pattern)
{

}
};
```

Java:

```
class Solution {
public int[] findPattern(int[][] board, String[] pattern) {

}
}
```

Python3:

```
class Solution:
def findPattern(self, board: List[List[int]], pattern: List[str]) ->
List[int]:
```