Problem 1128: Number of Equivalent Domino Pairs

Problem Information

Difficulty: Easy
Acceptance Rate: 60.58%
Paid Only: No
Tags: Array, Hash Table, Counting

Problem Description

Given a list of `dominoes`, `dominoes[i] = [a, b]` is **equivalent to** `dominoes[j] = [c, d]` if and only if either (`a == c` and `b == d`), or (`a == d` and `b == c`) - that is, one domino can be rotated to be equal to another domino.

Return _the number of pairs_`(i, j)`_for which_`0 <= i < j < dominoes.length` _, and_`dominoes[i]`_is**equivalent to** _`dominoes[j]`.

Example 1:

Input: dominoes = [[1,2],[2,1],[3,4],[5,6]] **Output:** 1

Example 2:

Input: dominoes = [[1,2],[1,2],[1,1],[1,2],[2,2]] **Output:** 3

Constraints:

* `1 <= dominoes.length <= 4 * 104` * `dominoes[i].length == 2` * `1 <= dominoes[i][j] <= 9`

Code Snippets

C++:

```
class Solution {
public:
int numEquivDominoPairs(vector<vector<int>>& dominoes) {

}
};
```

Java:

```
class Solution {
public int numEquivDominoPairs(int[][] dominoes) {

}
}
```

Python3:

```
class Solution:
def numEquivDominoPairs(self, dominoes: List[List[int]]) -> int:
```