**[Isomorphic Strings](https://leetcode.com/problems/isomorphic-strings/)**

Given two strings s and t, *determine if they are isomorphic*.

Two strings s and t are isomorphic if the characters in s can be replaced to get t.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

**Example 1:**

**Input:** s = "egg", t = "add"

**Output:** true

**Example 2:**

**Input:** s = "foo", t = "bar"

**Output:** false

**Example 3:**

**Input:** s = "paper", t = "title"

**Output:** true

**Constraints:**

* 1 <= s.length <= 5 \* 104
* t.length == s.length
* s and t consist of any valid ascii character.

class Solution {

public:

    bool isIsomorphic(string s, string t) {

        vector<int> indexS(200, 0); // Stores index of characters in string s

        vector<int> indexT(200, 0); // Stores index of characters in string t

        int len = s.length(); // Get the length of both strings

        if(len != t.length()) { // If the lengths of the two strings are different, they can't be isomorphic

            return false;

        }

        for(int i = 0; i < len; i++) { // Iterate through each character of the strings

            if(indexS[s[i]] != indexT[t[i]]) { // Check if the index of the current character in string s is different from the index of the corresponding character in string t

                return false; // If different, strings are not isomorphic

            }

            indexS[s[i]] = i + 1; // updating position of current character

            indexT[t[i]] = i + 1;

        }

        return true; // If the loop completes without returning false, strings are isomorphic

    }

};

Link : <https://leetcode.com/problems/isomorphic-strings/?envType=daily-question&envId=2024-04-02>