205 Isomorphic Strings (link)

Description

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 * 10^4`
- `t.length == s.length`
- `s` and `t` consist of any valid ascii character.

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <string>
#include <unordered_map>

using namespace std;

class Solution {
public:
    bool isIsomorphic(string s, string t) {
        unordered_map<char, char> charMapS;
        unordered_map<char, char> charMapT;

        int n = s.size();

        for (int i = 0; i < n; ++i) {
            char cs = s[i];
            char ct = t[i];

            if (charMapS.find(cs) != charMapS.end()) {
                if (charMapS[cs] != ct) {
                    return false;
                }
            } else {
                charMapS[cs] = ct;
            }

            if (charMapT.find(ct) != charMapT.end()) {
                if (charMapT[ct] != cs) {
                    return false;
                }
            } else {
                charMapT[ct] = cs;
            }
        }

        return true;
    }
};
```