

205. 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:

- $1 \leq s.length \leq 5 * 10^4$
 - `t.length == s.length`
 - `s` and `t` consist of any valid ascii character.
-

```

class Solution {
public:
    bool isIsomorphic(string str1, string str2) {

        if(str1.size() != str2.size()) return false;

        char match1[256];
            //fill the whole array with space
        memset(match1, ' ', sizeof(match1));

        for(int i = 0; i < str1.size(); i++) {
            if(match1[str1[i]] == ' ') match1[str1[i]] = str2[i];
            else {
                if(match1[str1[i]] != str2[i]) return false;
            }
        }
        //fill the whole array with space
        memset(match1, ' ', sizeof(match1));

        //we have to check both string matching because no string should have repeated characters matching with another character
        for(int i = 0; i < str2.size(); i++) {
            if(match1[str2[i]] == ' ') match1[str2[i]] = str1[i];
            else {
                if(match1[str2[i]] != str1[i]) return false;
            }
        }

        return true;
    }
};

```
