Isomorphic Strings

Thursday, December 16, 2021 2:23 PM

Isomorphic Strings

Easy Accuracy: 47.07% Submissions: 37831 Points: 2

Given two strings '**str1**' and '**str2**', check if these two strings are isomorphic to each other.
Two strings str1 and str2 are called isomorphic if there is a one to one mapping possible for every character of str1 to every character of str2 while **preserving the order**.
Note: All occurrences of every character in 'str1' should map to the same character in 'str2'

Example 1:

```
Input:
str1 = aab
str2 = xxy
Output: 1
Explanation: There are two different
charactersin aab and xxy, i.e a and b
with frequency 2and 1 respectively.
```

Example 2:

```
Input:
str1 = aab
str2 = xyz
Output: 0
Explanation: There are two different
different charactersin xyz. So there
won't be one to one mapping between
str1 and str2.
```

Your Task:
You don't need to read input or print anything.Your task is to complete the function **areIsomorphic**() which takes the string **str1** and string **str2** as input parameter and check if two strings are isomorphic. The function returns **true** if strings are isomorphic else it returns **false**.

Expected Time Complexity: O(|str1|+|str2|).
Expected Auxiliary Space: O(Number of different characters).
Note: |s| represents the length of string s.

Constraints:
1 <= |str1|, |str2| <= $2*10^4$

View Bookmarked Problems

Company Tags

View Bookmarked Problems

Company Tags

- Google

Topic Tags

- Strings

A **Simple Solution** is to consider every character of 'str1' and check if all occurrences of it map to the same character in 'str2'. The time complexity of this solution is O(n*n).

An **Efficient Solution** can solve this problem in O(n) time. The idea is to create an array to store mappings of processed characters.

```
1) If lengths of str1 and str2 are not same, return false.
2) Do following for every character in str1 and str2
a) If this character is seen first time in str1,
then current of str2 must have not appeared before.
(i) If current character of str2 is seen, return false.
Mark current character of str2 as visited.
(ii) Store mapping of current characters.
b) Else check if previous occurrence of str1[i] mapped
to same character.
```

Below is the implementation of above idea :

From <https://practice.geeksforgeeks.org/problems/isomorphic-strings-1587115620/1#>

```
packagedsaProblems;

importjava.util.HashMap;

importjava.util.Map;

publicclassIsomorphicString{

publicstaticbooleanareIsomorphic(Stringstr1,Stringstr2)
{
```

```
//Yourcodehere
Map<Character,Character>map=newHashMap<>();
intcount=0;
intm=str1.length();
intn=str2.length();
if(m!=n)
returnfalse;

for(inti=0;i<n;i++){
if(map.containsKey(str1.charAt(i)))
{
charvalue=str2.charAt(i);
if(value==map.get(str1.charAt(i)))
{
count++;
}
else
returnfalse;



}
else{
if(map.containsValue(str2.charAt(i)))
returnfalse;

else{
map.put(str1.charAt(i),str2.charAt(i));
count++;

}

}



}
if(count==m){
//System.out.println("Count:"+count);
returntrue;
}



else{
//System.out.println("InsideLastelse::Count:"+count);
returnfalse;
}
```

```
}
publicstaticvoidmain(String[]args){

Stringstr1="pijthbsfy";
Stringstr2="fvladzpbf";
booleanoutput=areIsomorphic(str1,str2);
System.out.println("Output:"+output);

}
}
```