/\*

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

Two strings 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.

For example,

Given "egg", "add", return true.

Given "foo", "bar", return false.

Given "paper", "title", return true.

法一：map<char,char>

\*/

class Solution {

public:

bool isIsomorphic(string s, string t)

{

if(s.size()!=t.size())

return false;

int n=s.size();

map<char,char> p;//s->t

map<char,char> g;//t->s

for(int i=0;i<n;i++)

{

map<char,char>::iterator it1=p.find(s[i]);

map<char,char>::iterator it2=g.find(t[i]);

if(it1==p.end() && it2==g.end())

{

p[s[i]]=t[i];

g[t[i]]=s[i];

}

else if(it1!=p.end())

{

if(it1->second!=t[i])

return 0;

}

else if(it2!=g.end())

{

if(it2->second!=s[i])

return 0;

}

}

return 1;

}

};