经过测试，我的代码不能满足题目的要求，我想知道问题出在哪

题目：

给你一个字符串 s 、一个字符串 t 。返回 s 中涵盖 t 所有字符的最小子串。如果 s 中不存在涵盖 t 所有字符的子串，则返回空字符串 "" 。对于 t 中重复字符，我们寻找的子字符串中该字符数量必须不少于 t 中该字符数量。如果 s 中存在这样的子串，我们保证它是唯一的答案。

我的代码：

#include<unordered\_map>

bool ishave(unordered\_map<char,int> father,unordered\_map<char,int> son){

//判断father中是否含有完整的son

unordered\_map<char,int>::iterator it = son.begin();

for(;it!=son.end();++it){

if(father.find(it->first)==father.end()) return 0;

else{

if(father.find(it->first)->second<it->second) return 0;

}

}

return 1;

}

class Solution {

public:

string minWindow(string s, string t) {

unordered\_map<char,int> son;

for(int i=0;i<t.size();++i) son[t[i]]++;//子串

int minlen=INT32\_MAX;//记录最短长度

unordered\_map<char,int> father;//待匹配子串

int left=0,minleft=0;//子串起点

int right=0;//子串下一个

father[s[0]]++;//初始子串

if(ishave(father,son)) minlen=1;

while(right<s.size()){

father[right++]++;//加入新元素

while(ishave(father,son)){//满足条件的处理

if(right-left<minlen){

minlen=right-left;

minleft=left;

}

father[s[left++]]--;//删除第一个元素

if(father.find(s[left-1])->second==0) father.erase(s[left-1]);

}

}

if(minlen==INT32\_MAX) return "";

else return s.substr(minleft,minlen);

};

};