源程序的相似性

问题描述

对于两个C++语言的源程序代码，用哈希表的方法分别统计两个程序中使用C++语言关键字的情况，并最终按定量的计算结果，得出两份程序的相似性。

基本要求

建立C++语言关键字的哈希表，统计在每个源程序中C++关键字出现的频度, 得到两个向量X1和X2，通过计算向量X1和X2的相对距离来判断两个源程序的相似性。   
例如:    
    关键字        Void  Int  For  Char  if  else  while  do  break  class   
程序1关键字频度    4    3      0    4    3    0    7    0    0      2   
程序2关键字频度    4    2      0    5    4    0    5    2    0      1   
    X1=[4,3,0,4,3,0,7,0,0,2]   
 X2=[4,2,0,5,4,0,5,2,0,1]   
    设s是向量X1和X2的相对距离，s=sqrt( ∑(x1[i]-x2[i]) 2 )，当X1=X2时，s=0, 反映出可能是同一个程序；s值越大，则两个程序的差别可能也越大。

**测试数据**

选择若干组编译和运行都无误的C++程序，程序之间有相近的和差别大的，用上述方法求s, 对比两个程序的相似性。

选作内容

建立源代码用户标识符表，比较两个源代码用户标识符出现的频度，综合关键字频度和用户标识符频度判断两个程序的相似性。

一、**实验目的**

1.熟悉C++语言的上机环境，进一步掌握C++语言的应用。

2.熟悉哈希表的结构，性质；掌握用c++语言实现哈希表的建立，使用方法。

3.解决判断两份程序的相似性问题，增强解决实际问题的能力。

## 二、实验内容

## 对于两个C++语言的源程序代码，用哈希表的方法分别统计两个程序中C++语言关键字的情况，并最终按定量的计算结果，得出两份程序的相似性。

## 三、实验步骤

## 1. 建立C++语言关键字的哈希表，设置关键字内容。

## 2.读取程序所在文件内容，统计关键字出现频度，使用 哈希表记录统计结果。

## 3. 由每个源程序中C++关键字出现的频度, 得到两个向量X1和X2，计算向量X1和X2的相对距离来判断两个源程序的相似性

## 四、完整参考程序

## #include<iostream>

## #include<fstream>

## #include<cmath>

## using namespace std;

## #define max 10 //哈希表长

## 

## class Hash{ //哈希表类

## public:

## string \*hash\_str; //哈希表关键字

## int hash\_X1[max]={0}; //程序1关键字频度

## int hash\_X2[max]={0}; //程序2关键字频度

## Hash(string str[]){ //构造函数，赋值哈希表关键字

## hash\_str = str;

## }

## ~Hash(){} //析构函数

## };

## int main()

## {

## string str[max] = {"void","int","for","char","if","else","while","do","break","class"};

## Hash\* hash = new Hash(str); //实例化哈希表 ，传入关键字数组

## 

## 

## ifstream fin\_X1("file\_in\_1.txt"); //读取程序1

## ifstream fin\_X2("file\_in\_2.txt"); //读取程序2

## string word;

## 

## 

## while(fin\_X1>>word){ //逐个读取字符串

## for(int i=0;i<max;i++){

## if(word==hash->hash\_str[i]) hash->hash\_X1[i]++; //用哈希表记录关键字出现频度

## }

## }

## 

## while(fin\_X2>>word){

## for(int i=0;i<max;i++){

## if(word==hash->hash\_str[i]) hash->hash\_X2[i]++;

## 

## }

## }

## 

## cout<<"关键字："<<'\t'<<'\t';

## for(int j=0;j<max;j++){ //输出关键字

## cout<<hash->hash\_str[j]<<'\t';

## }

## cout<<endl<<endl<<"程序1关键字频度:"<<'\t';

## for(int j=0;j<max;j++){ //输出程序1关键字频度

## cout<<hash->hash\_X1[j]<<'\t';

## }

## cout<<endl<<endl<<"程序2关键字频度:"<<'\t';

## for(int j=0;j<max;j++){ //输出程序2关键字频度

## cout<<hash->hash\_X2[j]<<'\t';

## }

## fin\_X1.close(); //关闭文件读取

## fin\_X2.close();

## 

## double s=0; //计算相似度 s

## for(int k=0;k<10;k++){

## s+=(hash->hash\_X1[k]-hash->hash\_X2[k])\*(hash->hash\_X1[k]-hash->hash\_X2[k]);

## }

## s=sqrt(s);

## cout<<endl<<endl<<"相似度："<<fixed<<s<<endl;

## return 0;

## }