基于例2.6的Kasiski测试法编程

20215223 杨佳俊

1. **源程序代码**

#include <iostream>

#include <vector>

#include <iomanip>

#include <cmath>

using namespace std;

int main() {

// 密文字符串

string ciphertext =

"CHREEVOAHMAERATBIAXXWTNXBEEOPHBSBQMQEQERBW"

"RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAX"

"LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX"

"VRVPRTULHDNQWTWDTYGBPHXTFALJHASVBFXNGLLCHR"

"ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT"

"AMRVLCRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI"

"PEEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP"

"WQAIIWXNRMGWOIIFKEE";

// 每组字符串的长度

int n = ciphertext.length() / 5 + 1;

// 初始化二维数组，存储每组中各个字符的出现次数

vector<vector<int>> charCounts(5, vector<int>(26, 0));

// 遍历密文，分组并统计字符A-Z出现次数

for (int i = 0; i < ciphertext.length(); ++i) {

int group = i % 5;

if (isalpha(ciphertext[i])) {

charCounts[group][ciphertext[i] - 'A']++;

}

}

// 初始化26个英文字母的出现频率

float pi[26] = { 0.082, 0.015, 0.028, 0.043, 0.127, 0.022, 0.020, 0.061, 0.070, 0.002,

0.008, 0.040, 0.024, 0.067, 0.075, 0.019, 0.001, 0.060, 0.063, 0.091,

0.028, 0.010, 0.023, 0.001, 0.020, 0.001 };

// 输出统计结果

cout << "统计结果：" << endl;

cout << "A 到 Z：";

for (char ch = 'A'; ch <= 'Z'; ++ch) {

cout << ch << " ";

}

cout << endl;

for (int group = 0; group < 5; ++group) {

cout << "第" << group + 1 << "组：";

for (int i = 0; i < 26; ++i) {

cout << " " << charCounts[group][i];

}

cout << endl;

}

// m数组用于存储Mg的值

float m[5][26];

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 26; j++) {

m[i][j] = 0; // 初始化为零

}

vector<vector<int>> tempCharCounts(charCounts); // 复制一个辅助数组，用于存储每次移位后的数组

for (int shiftAmount = 0; shiftAmount < 26; shiftAmount++) {

for (int k = 0; k < 26; k++) {

charCounts[i][(k + 26 - shiftAmount) % 26] = tempCharCounts[i][k];

}

for (int j = 0; j < 26; j++) {

m[i][shiftAmount] += pi[j] \* charCounts[i][j] / n; // 结果进行累加

}

}

}

// 输出 m 数组

cout << "\nm 数组：" << endl;

for (int i = 0; i < 5; i++) {

cout << "第" << i + 1 << "组：" << endl;

for (int j = 0; j < 26; j++) {

// 用于将结果保留三位小数输出

// cout << " " << fixed << setprecision(3) << round(m[i][j] \* 1000) / 1000;

cout << " " << m[i][j];

if (j % 10 == 9) {

cout << endl;

}

}

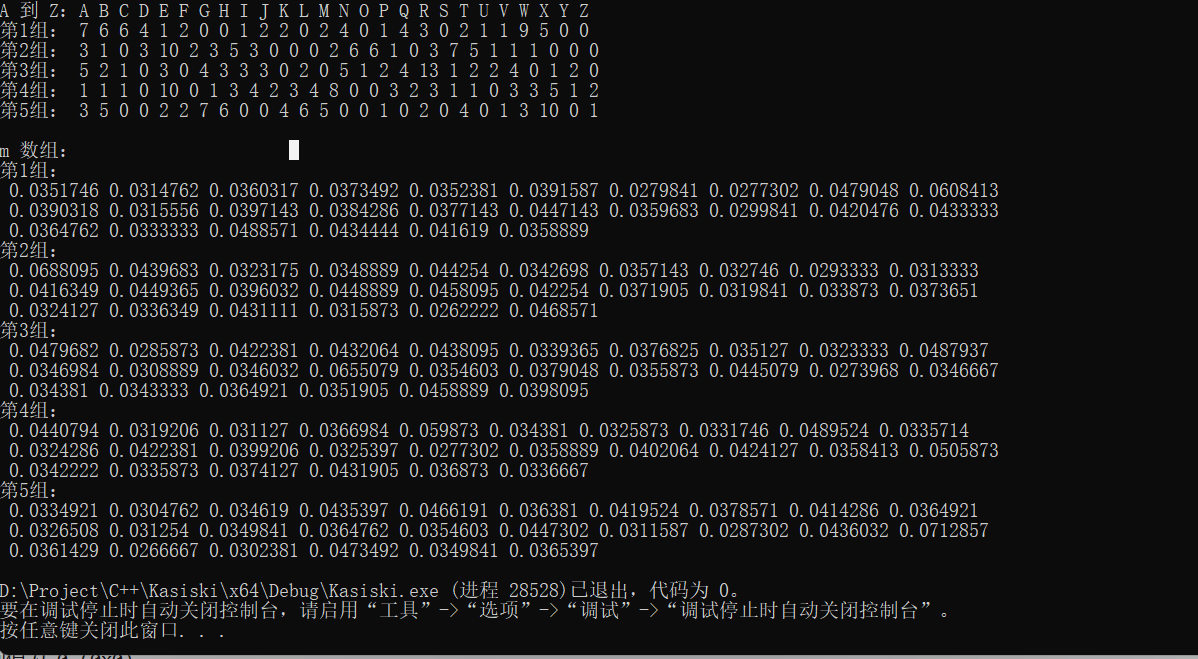
cout << endl;

}

return 0;

}

1. **实验结果**



上方显示的是分组后各字母的统计频次

下方显示的是计算出的Mg结果，下图是保留三位小数结果

结果与课件上一致

