

学生实验报告



课程名称\_\_\_数据结构基础\_

题目名称\_\_\_请输入实验题目\_\_

小组成员 31901144 石子杰

（学号-姓名）

31901145 史吴嘉豪

31901146 孙照浩

任课教师\_\_ 王云武 \_

2020 年6月 7 日

# 零.完成情况

**PINTIA评测成绩：100**

**成员分工及贡献情况：**石子杰 30% 史吴嘉豪40% 孙照浩 30%

# 问题描述

题目：字符串匹配

给出一个最大长度为10^6的母串t，请你在t里面找到长度为len的出现次数最多的子串，如果找到多个出现次数一样的子串，请你输出字典序最小的。

**输入格式**:

在第一行输入一个正整数Len（Len<=10^6），第二行输入一个母串t，t的长度小于等于10^6。

**输出格式**:

输出答案子串和它在t中的出现次数，用一个空格分隔，行末尾没有多余空格！

**输入样例:**

在这里给出一组输入。例如：

3

aba ababababababaaababababa

**输出样例:**

在这里给出相应的输出。例如：

aba 11

# 二.数据结构定义与分析

创建键值对map<string,int>result，其中我们要输入的字符串是主键。

创建动态数组（迭代器）map<string,int>::iterator iter，iter指向字符串中的逐个字符。iter->first为string类型，iter->second为int类型变量。对应map中Result\_string和Result\_number。

# 三、算法分析

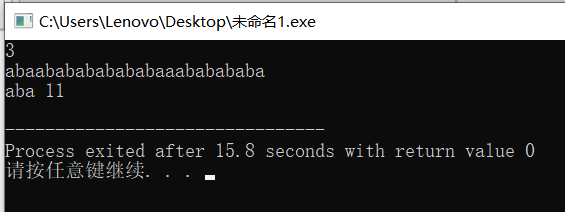
在声明要用到的键值对和迭代器后，依次输入要截取的字符串长度len和字符串。然后通过for语句循环遍历整个字符串，截取长度为len的子串存储在map中，主键是截取到的字符串的迭代器的int值+1。然后返回指向map头部的迭代器，将头部迭代器的int值赋给 Result\_number 变量，将头部迭代器的string值赋给 Result\_string 变量，这就将Result\_string当作当前出现次数最多的子串了，将Result\_number当作该子串出现次数。然后用while循环，直到迭代器指向末端元素（末字符）为止，对迭代器的值进行遍历。如果当前迭代器中的 int 值大于 Result\_number 变量的值，就将当前迭代器的int值赋给 Result\_number 变量，将当前迭代器的string值赋给 Result\_string 变量，这样就更新了出现次数最多的子串和他的出现次数。程序的最后输出当前的Result\_string和Result\_number即可。

# 四. 算法的时间复杂度分析

算法的时间复杂度：O(n),因为要充分截取每一个字符，不遗漏任何一个子串，所以需要一个个遍历。

# 五、测试用例的选择原因以及测试结果

用abaababababababaaababababa这样的测试用例，尽量做到复杂性，其中存在大量的覆叠在一起的aba，能够较好地测试程序的适用性。bab也重复出现多次，测试了题目要求的“输出出现最多的字串”的比较功能。

测试结果：

# 六、存在问题及可能原因

cin可能会导致读取字符串时出现一些问题。

# 七、附源代码以及代码相应注释

#include<bits/stdc++.h>

using namespace std;

map<string,int>result; //创建一个以字符串值为主键的map

map<string,int>::iterator iter; //声明一个迭代器

int main(){

int Len,i,Result\_number;

cin >> Len; //读取长度

string t,Temp,Result\_string;

cin >> t; //读取字符串

for (i=0;i<=t.length()-Len;i++)

{

Temp=t.substr(i,Len); //截取字符串

result[Temp]++; //在map中主键为截取到的字符串的迭代器的int值+1

}

iter=result.begin(); //返回指向map头部的迭代器

Result\_number = iter->second; //将头部迭代器的int值赋给 Result\_number 变量

Result\_string = iter->first; //将头部迭代器的string值赋给 Result\_string 变量

while(iter != result.end())

{

if (iter->second > Result\_number) //当前迭代器中的 int 值大于 Result\_number 变量的值时

{

Result\_number = iter ->second; //将当前迭代器的int值赋给 Result\_number 变量

Result\_string = iter->first; //将当前迭代器的string值赋给 Result\_string 变量

}

iter++;

}

cout << Result\_string << ' ' << Result\_number << endl; //输出

return 0;

}

# 八、实验创新点或合理化建议

使用了C++标准模板库中的map模板，简化了程序编写的过程，将手动构造散列函数和处理冲突的过程简化为调用map模板自动处理，提高了编写程序的效率。