**算法分析与设计实验报告**

**第 1 次实验**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 杨杰 | 学号 | 201908010705 | | 班级 | 计科1907 |
| 时间 | 3.19 | 地点 | 软件大楼 | | | |
| 实验名称 | 字典序问题 | | | | | |
| 实验目的 | 通过上机实验，要求掌握字典序问题的问题描述、算法设计思想、程序设计。 | | | | | |
| 实验原理 | 查找字符串在字典中的编码，根据不同数据规模的测试用例，能准确输出测试用例字符串在字典中的编码，并计算出程序运行所需要的时间。 | | | | | |
| 实验步骤 | * + - 1. 进行数学推导，设计需要哪些函数来实现相应功能。  1. 构造组合数com函数，用来求的值。 2. 构造num函数，用来计算从以a字母打头，到以b字母打头，长度为l的升序字符串个数。 3. 最后构造fun函数，计算给定字符串在字典中的字典序。 | | | | | |
| 关键代码 | int com(int m,int n)*//计算组合数*  {      int result=1;      for(int i=0;i<n;i++)          result=result\*(m-i)/(i+1);      return result;  }  int num(char a,char b,int l)*//计算从以a字母打头，到以b字母打头，长度为l的升序字符串个数*  {      int sum=0;      for(int i=a;i<=b;i++)          sum+=com(122-i,l-1);      return sum;  }  void fun(string s)*//计算给定字符串在字典中的字典序*  {      int len=s.length();      int sum=0;      for(int i=1;i<len;i++)          sum+=num('a',char(123-i),i);      sum+=num('a',s[0]-1,len);      int k=1;      for(int i=len-1;i>0;i--)      {              sum+=num(s[k-1]+1,s[k]-1,i);*//*              k++;      }      outfile<<"计算字符串"<<s<<"的字典序,结果为"<<sum+1<<endl;  } | | | | | |
| 算法复杂度分析 | 先逐个函数进行分析，com函数一层循环，时间复杂度为O(len);num函数二层循环，时间复杂度为O((b-a)\*len); fun函数三层循环，时间复杂度为O((b-a)\*len2);  在main函数中，四层循环，时间复杂度为O(n\*(b-a)\*len2). | | | | | |
| 测试结果  （含运行时间） |  | | | | | |
| 类似题目 | 本题中要求长度小的字符串排在前面，还有一类题目是要求按照字典序排列，如a<ab<ac<acd<ad,解决这类题目需要将关键代码修改如下：  int com(int m,int n)*//计算组合数*  {      int result=1;      for(int i=0;i<n;i++)          result=result\*(m-i)/(i+1);      return result;  }  int num(char a)*//计算以a字母打头，长度从1到最大的升序字符串个数*  {      int sum=0;      for(int i=1;i<=123-a;i++)          sum+=com(122-a,i-1);      return sum;  }  void fun(string s)*//计算给定字符串在字典中的字典序*  {      int len=s.length();      int sum=0;      for(char ch='a';ch<s[0];ch++)          sum+=num(ch);      for(int i=1;i<len;i++)          for(char ch=s[i-1]+1;ch<s[i];ch++)              sum+=num(ch);      outfile<<"计算字符串"<<s<<"的字典序,结果为"<<sum+len<<endl;  }  测试结果为： | | | | | |
| 实验心得 | 在编写组合数函数时，最开始想通过数学公式=m!/(n!\*(m-n)!)来实现，但是这样有个问题，那就是n！增长速度是非常快的，哪怕是long long的数据类型也很容易溢出，同时阶乘函数要用到递归，空间开销也很大，所以对算法进行改进，解决了上述问题。  在fun函数中有这样一条语句：  *sum+=num(s[k-1]+1,s[k++]-1,i);*  它的本意是计算从前一位字符的下一个字符打头，到当前位字符的上一个字符打头，长度为i的升序字符串的个数，但是在实际操作中发现，这样总是会出现错误，例如对于字符串bdf，k初始值为1，i初始值为2，函数传进去的参数应为c,c,2，但实际上却是e,c,2.这是因为在原代码中k和k++同时作为参数传递，不同编译器对此有不同理解，为避免造成歧义，将代码修正如下：              sum+=num(s[k-1]+1,s[k]-1,i);  *//sum+=num(s[k-1]+1,s[k++]-1,i);*              k++;  经过这次实验，我对于字典序问题的相关代码已基本熟悉，算法知识得到了复习与巩固。在写代码与调试的过程中，在解决问题过程中，丰富了个人编程的经历和经验，提高了个人解决问题的能力。 | | | | | |
| 实验得分 |  | 助教签名 | |  | | |

**附录：完整代码**

*//字典序问题*

#include <iostream>

#include <fstream>

#include <string.h>

#include <windows.h>

using namespace std;

ofstream outfile("zidianxu\_out.txt",ios::out);

int com(int m,int n)*//计算组合数*

{

    int result=1;

    for(int i=0;i<n;i++)

        result=result\*(m-i)/(i+1);

    return result;

}

int num(char a,char b,int l)*//计算从以a字母打头，到以b字母打头，长度为l的升序字符串个数*

{

    int sum=0;

    for(int i=a;i<=b;i++)

        sum+=com(122-i,l-1);

    return sum;

}

void fun(string s)*//计算给定字符串在字典中的字典序*

{

    int len=s.length();

    int sum=0;

    for(int i=1;i<len;i++)

        sum+=num('a',char(123-i),i);

    sum+=num('a',s[0]-1,len);

    int k=1;

    for(int i=len-1;i>0;i--)

    {

            sum+=num(s[k-1]+1,s[k]-1,i);*//*

            k++;

    }

    outfile<<"计算字符串"<<s<<"的字典序,结果为"<<sum+1<<endl;

}

int main()

{

    ifstream infile("zidianxu\_in.txt",ios::in);

    int x=0;

    infile>>x;

    string str;

    double time=0;

    LARGE\_INTEGER nFreq,nBeginTime,nEndTime;

    while(infile>>str)

    {

        QueryPerformanceFrequency(&nFreq);

        QueryPerformanceCounter(&nBeginTime);*//开始计时*

        fun(str);

        QueryPerformanceCounter(&nEndTime);*//停止计时*

        time=(double)(nEndTime.QuadPart-nBeginTime.QuadPart)/(double)nFreq.QuadPart;*//计算程序执行时间单位为s*

        outfile<<"运行时间为"<<time\*1000<<"ms"<<endl;

    }

    infile.close();

    outfile.close();

    return 0;

}

*//字典序数据生成器*

#include <iostream>

#include <fstream>

#include <time.h>

#define random(a,b) (rand()%(b-a+1)+a)

using namespace std;

int main()

{

    int t=0;

    cin>>t;

    ofstream outfile("zidianxu\_in.txt",ios::out);

    outfile<<t<<endl;

    srand((int)time(NULL));

    for(int i=0;i<t;i++)

    {

        int l=random(1,26);

        outfile<<l<<' ';

        int k=random(97,122);

        outfile<<char(k);

        for(int j=1;j<l;j++)

        {

            if(k==122) break;

            k=random(k+1,122);

            outfile<<char(k);

        }

        outfile<<endl;

    }

    outfile.close();

    return 0;

}