**暨南大学本科实验报告专用纸**

课程名称 算法分析与设计实验 成绩评定

实验项目名称 指导教师 李展 实验地点 N116

实验项目编号 001 实验项目类型 综合型

学生姓名 张瑞鹏 学号 2020101124

学院 信息科学技术学院 系 计算机 专业 计算机科学与技术

实验时间 2022 年 月 日 午～ 月 日 下 午

1. **问题描述**

**问题：一本书的页码从自然数1开始顺序编码直到自然数n。求的页码按照通常的习惯编排,每个页码都不含多余的前...**

**一本书的页码从自然数1开始顺序编码直到自然数n。求的页码按照通常的习惯编排,每个页码都不含多余的前导数字0。例如第6页为6而非06或006等。试统计给定书的总页码n，计算出书的全部页码中分别用到多少次数字0,1,2…,9。**

1. **算法思路**（用文字简单说明）

方法一：对每一位数求使用的次数

假设数组是：x0x1x2x3x4x5

考虑个位所出现的数：

首先mod10取出个位的数字x5, x0x1x2x3x4x5, x0x1x2x3x4(x5-1)……x0x1x2x3x4 0考虑到这几个数，仅对于个位来说，0~x5的数字使用次数+=1；

此时，只需考虑比x0x1x2x3x4小的所有数组在个位数字变化的情况，就能完全枚举出个位出现过什么数字。个位变化一轮（即0~9），则高位进一，因此可以看作进位x0x1x2x3x4次。因此可以看作（0~9）的使用次数+= x0x1x2x3x4。

注意：此时算了0，所以有a[0]—即可；

此时个位情况考虑完毕。

对于最高位：

取出最高位x0,先计算x0 00000~ x0 x1x2x3x4x5,最高位的情况，可以看到这里一共有x1x2x3x4x5+1个数字，因此x0的次数+= x1x2x3x4x5+1；

计算完成之后只需要考虑比x0 00000小的数字的情况，（100000~（x0-1）99999）,一共有这么多个数，可以看到1~（x0-1）个数，每个数均出现了100000次，例如（100000~199999），出现多少次具体还需要看最高位处在第几位。

此时最高位的情况考虑完毕。

对于其他位数：

对于其他的位数，大致相当于上述两种情况的综合。

首先，取出当前位置，假设取出x3（即百位上）

先向低位考虑，末尾有x4x5。最高位的计算，x3 += x4x5+1。然后对（0~x3-1）全体+=100（这个100根据位置决定），

再向高位考虑，类似与最低位置。由于已经处理了上面的情况，只需考虑比x0x1x2 000的所有情况，可以看作（1~x0x1x2000）,百位经历一轮（0~9）则向高位进位，高位进位一共是x0x1x2,0~9全部+=x0x1x2;（注意：同理，a[0]-=position;)

遍历所有中间位数，这样就完成所有位数数字使用次数的求和。

方法二：

遍历所页数，对于每次的页数，依次mod10取余数。取出来的余数++，遍历完成之后便得到结果。

**（三）算法实施步骤和流程**（伪代码/流程图等方式描述）

**方法一：**

**处理最低为的算法：**

**int single = num % 10;//取出个位数**

**int left = num / 10;//取出其余位数**

**for (int i = 0; i <= single; i++){**

**a[i] += 1;//**

**}//各位多余的数字**

**for (int i = 0; i <= 9; i++) {**

**a[i] += left;**

**}**

**a[0] -= 1;**

**处理中间位数的算法**

**for (int i = 2; i <= len - 1; i++){**

**int position = 1;**

**for (int j = 0; j <i-1; j++){**

**position \*= 10;**

**}**

**int rightnow = (num / position) % 10;//取出当前位数。这个和处理的位数有关**

**int before = num % position;//取出在此之前的数，这个和处理的位数有关**

**int after = num / (position\*10);//取出在此之后的数字**

**a[rightnow] += before + 1;**

**for (int i = 0; i < rightnow; i++){**

**a[i] += position;//这个和处理的位数有关**

**}//处理多出来的数字**

**for (int i = 0; i <= 9; i++){**

**a[i] += after \* position;**

**}//算当前数字的一轮**

**a[0]-=position;**

**}**

**处理最高位的算法：**

**int position = 1;**

**for (int i = 0; i < len - 1; i++){**

**position \*= 10;**

**}//用于确定是是哪一个数位**

**int rightnow = num / position;//取出当前位数。这个和处理的位数有关**

**int before = num % position;//取出在此之前的数，这个和处理的位数有关**

**a[rightnow] += before + 1;**

**printf("打印现在数值大小%d\n", rightnow);**

**for(int i = 1; i < rightnow; i++){**

**a[i] += position;//这个和处理的位数有关**

**}//处理多出来的数字**

**方法二：**

**for (i = 1; i <= n; i++)//逐页拆数字的方法 {**

**int k=i;**

**while(k){**

**a[k%10]++;**

**k = k/10;**

**}**

**}**

**（四）源代码**（通过了编译运行的正确程序）

**#include<stdio.h>**

**#include<iostream>**

**#include<string.h>**

**using namespace std;**

**void transfrom(char\* text,long &num)//字符串模式转换为整型模式**

**{**

**num = 0;**

**int len = strlen(text);**

**for(int i =0;i<len;i++)**

**{**

**int position = 1;**

**for(int j = 0;j<len-i-1;j++){**

**position\*=10;**

**}**

**printf("展示大小%d\n",text[i]-'0');**

**num=num+((int)(text[i]-'0'))\*position;**

**}**

**printf("输出num的大小%d\n",num); printf("输出len的大小%d",len);**

**}**

**int main(void)**

**{**

**char text[10];//读入文件**

**FILE\* fp1;**

**fp1 = fopen("numberfile.txt","r");**

**fscanf(fp1,"%s",text);**

**fclose(fp1);//scanf("%s",text);**

**printf("测试数据是多少：%s\n",text);**

**long num;**

**transfrom(text,num);**

**int len = strlen(text);**

**//printf("%s\n",text);**

**//算法处理**

**long a[10] = { 0 };//初始化数组**

**//先算个位**

**int single = num % 10;//取出个位数**

**int left = num / 10;//取出其余位数**

**for (int i = 0; i <= single; i++)**

**{**

**a[i] += 1;//**

**}//各位多余的数字**

**for (int i = 0; i <= 9; i++) {**

**a[i] += left;**

**}**

**a[0] -= 1;**

**//printf("%d", a[0]);**

**//再算其他位数**

**for (int i = 2; i <= len - 1; i++)**

**{**

**int position = 1;**

**for (int j = 0; j <i-1; j++){**

**position \*= 10;**

**}**

**int rightnow = (num / position) % 10;//取出当前位数。这个和处理的位数有关**

**int before = num % position;//取出在此之前的数，这个和处理的位数有关**

**int after = num / (position\*10);//取出在此之后的数字**

**a[rightnow] += before + 1;**

**for (int i = 0; i < rightnow; i++){**

**a[i] += position;//这个和处理的位数有关**

**}//处理多出来的数字**

**for (int i = 0; i <= 9; i++){**

**a[i] += after \* position;**

**}//算当前数字的一轮**

**a[0]-=position;**

**}**

**//再算头顶位数**

**int position = 1;**

**for (int i = 0; i < len - 1; i++){**

**position \*= 10;**

**}**

**int rightnow = num / position;//取出当前位数。这个和处理的位数有关**

**int before = num % position;//取出在此之前的数，这个和处理的位数有关**

**a[rightnow] += before + 1;**

**printf("打印现在数值大小%d\n", rightnow);**

**for(int i = 1; i < rightnow; i++){**

**a[i] += position;//这个和处理的位数有关**

**}//处理多出来的数字**

**//输出文件**

**FILE\* fp2;fp2 = fopen("result.txt","w");**

**for (int i = 0; i <= 9; i++)**

**{**

**fprintf(fp2,"数字%d使用的次数是%d\n", i, a[i]);**

**}**

**fclose(fp2);**

**for (int i = 0; i <= 9; i++)**

**{**

**printf("数字%d使用的次数是%d\n", i, a[i]);**

**}**

**return 0;**

**}**

**方法二：**

**for (i = 1; i <= n; i++)//逐页拆数字的方法**

**{**

**int k=i;**

**while(k)**

**{**

**a[k%10]++;**

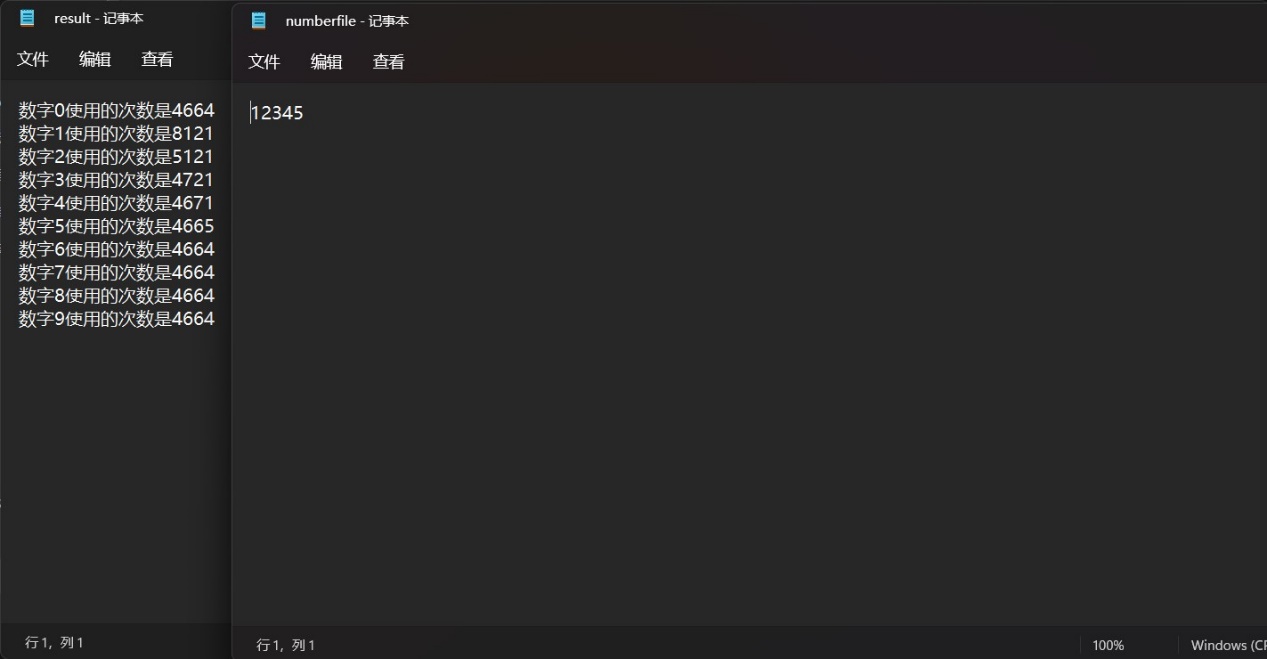
**k = k/10;**

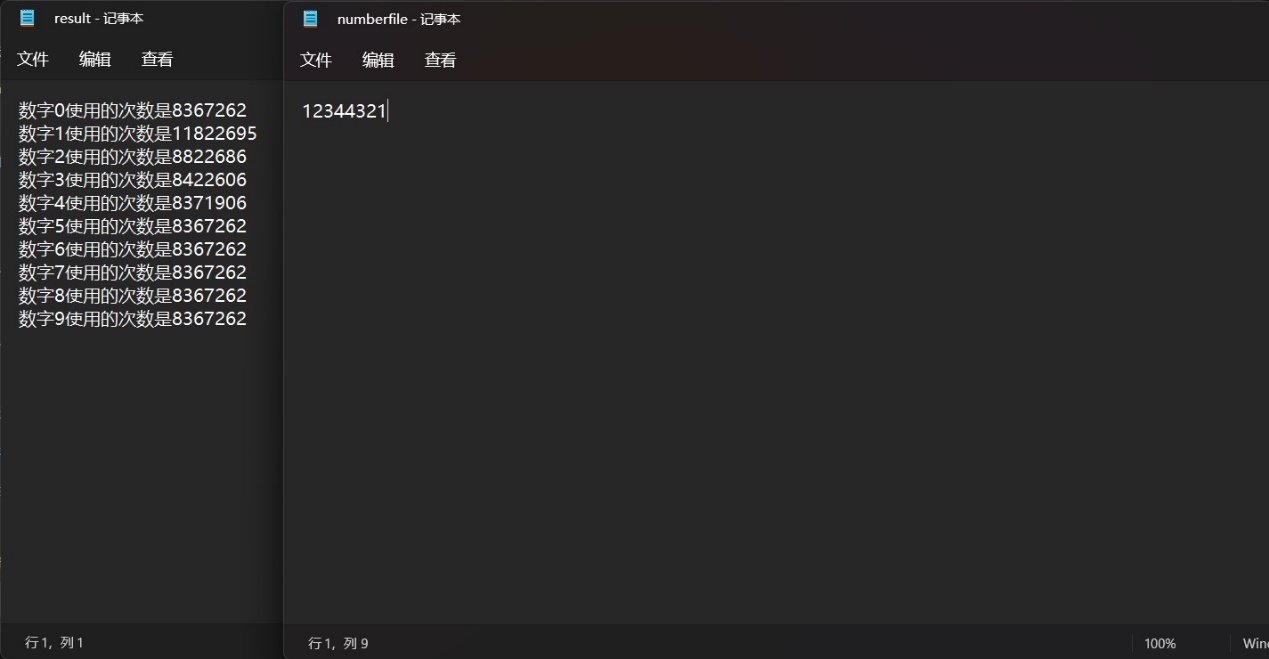
**}**

**}**

**算法部分改成这样，其他文件开关不变。**

**（五）测试结果**（至少有两个以上算例及程序运行结果，截图贴进实验报告）

****

****

**（六）实验总结**（至少三句话，可以写复杂度分析、遇到问题、可能的改进措施、心得体会等）

1.方法一的时间复杂度：

计算个位数的时间复杂度，只与个位数大小有关系，个位数大小是随机数，因此这一步于输入规模n无关

同理，计算最高位也是如此

计算中间位数的次数与输入长度有关，有等式C10^len = n(存在常数乘以10^len等于输入规模n)，化简得len = Clg10,因此时间与lg10有关

综上所述：时间复杂度为O(lg10)

2.方法二的时间复杂度为O（n）

3.中途遇关于0开头被算进去的问题，后来发现并解决。

4.自己设计的算法虽然代码很长，但是有成就感。

**暨南大学本科实验报告专用纸(附页)**