

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 统计数字问题 指导教师 李展
实验项目编号 实验一 实验项目类型 综合性 实验地点
学生姓名 张印祺 学号 2018051948
学院 信息科学技术 系 计算机科学 专业 网络工程
实验时间 2020 年 3 月 11 日下午 ~ 3 月 11 日 下 午

一、问题描述

给定表示书的总页码的十进制整数 n ($1 \leq n \leq 10^9$), 计算书的全部页码中分别用到多少次数数字 0, 1, 2, ..., 9。

一本书的页码从自然数 1 开始顺序编码直到自然数 n 。书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。计算出书的全部页码中分别用到多少次 0, 1, ..., 9。

二、算法思路

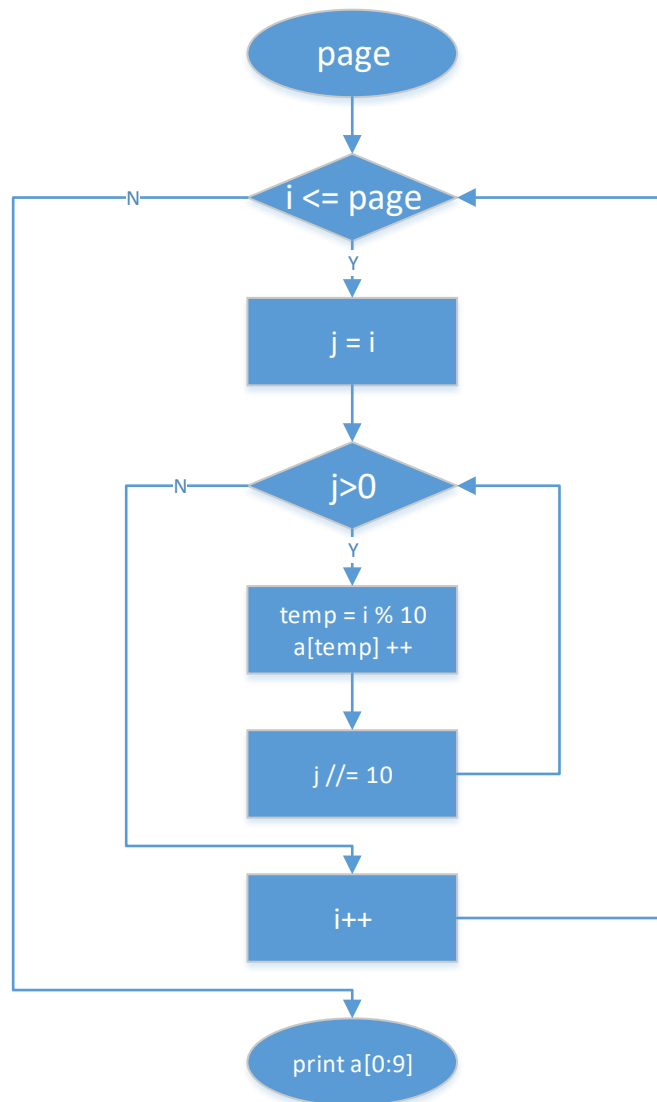
步骤一: 当输入一个数 n 时, 计算 1 到 n 每个数字中 1 到 0 出现的次数总和。

步骤二: 对于任意一个数字 n , 先取其个位数字 ($i = n \% 10$), 列表 `list[i]` 的记录内容加 1;

步骤三: n 取完个位数后向右移动一位 ($n /= 10$), 重复步骤二, 直到 n 小于等于 0 为止;

步骤四: 输出记录列表 `list` (保存到 `CountNum.txt`)。

三、算法实施流程



四、主要仪器设备

Win10 专业版
Visual Studio Code
Python3.7

五、测试结果

页码: 12345,	页码: 7123471,
0: 4664,	0: 4258987,
1: 8121,	1: 5293570,
2: 5121,	2: 5263569,
3: 4721,	3: 5259569,
4: 4671,	4: 5259069,
5: 4665,	5: 5258997,
6: 4664,	6: 5258997,
7: 4664,	7: 4382461,
8: 4664,	8: 4258987,
9: 4664	9: 4258987

五、实验总结

本算法主要两个循环步骤：

第一步，从 1 到 n 进行迭代。

第二步，对每一个数进行逐个计数。

显而易见，第一步的算法时间复杂度有渐进表达式 $T(n) = n$ ；

每一个数字的位数有公式：

$$b(i) = \text{Integer}(\log(i)) + 1$$

因此，第二步的时间复杂度有渐进表达式 $T(n) = \log n$ 。

在完整的算法运行时，我们假设每一条语句运行的时间为 k_i ，则内层循环每次运行时间与 i 的位数 $b(i)$ 正相关，程序运行的总时间就为 $\sum_{i=1}^n k_i + b(i)$ 。

$$\sum_{i=1}^n k_i = an, \quad \lim_{n \rightarrow \infty} \sum_{i=1}^n b(i) = n + \lg(n!), \text{ 由 } \textit{Stirling} \text{ 公式,}$$

$$\lim_{n \rightarrow \infty} \lg(n!) = \lim_{n \rightarrow \infty} n \lg(n)$$

综上，在 n 非常大时程序总的运行时间为 $n(1+a+\lg n)$ 。

$$\lim_{n \rightarrow \infty} \frac{n \lg n}{n} = \infty, \text{ 故算法的时间复杂度的渐进表达式 } T(n) = n \log n.$$

通过这次的实验，我将算法分析方法学以致用，巩固了认知。通过动手写程序增加了对程序语言的熟练度。同时在查阅资料的过程中收获了新的数学知识。

算法改进：

按每一位对数字进行统计。

举个例子，对于 12345，我们要统计 1 到 12345 中 1 的个数，我们先统计个位中可能出现 1 的情况，那么有 1234 种，统计十位数的 1，一共有 $123 * 10$ 种（abc1X，abc 共 123 种组合，x 有 10 种情况）；以此类推，对于 1 的统计有：

$$1234 * 10^0 + 123 * 10^1 + 12 * 10^2 + 1 * 10^3$$

运用这种算法，时间复杂度为 $\log n$ 。

六、源代码

#使用文件形式输入输出将注释内容删除

```
import os
```

```
PATH = ".\\\"
```

```
class Pages:
```

```
    def getPage(page):
```

```
        page = int(input("请输入页码"))#文件输入将其注释
```

```
        #f1 = open(PATH + "1.CountNum", "r", encoding = 'UTF-8')
```

```
        #page = int(f1.read())
```

```
        return page
```

```
    def CountNum(self, page):
```

```

        list = [0] * 10
        for i in range(1, page + 1):
            while i:
                a = (i % 10)
                list[a] += 1
                i //= 10
            return list
    pages = Pages()
    page1 = pages.getPage()
    list = pages.CountNum(page1)
    result = " 页 码 :  {},\n0:{},\n1:{},\n2:{},\n3:{},\n4:{},\n5:{} ,
\n6:{},\n7:{},\n8:{},\n9:{}".format(page1, list[0], list[1], list[2]
, list[3], list[4], list[5], list[6], list[7], list[8], list[9] )
    #f = open(PATH + '1.CountNum', 'a', encoding='UTF-8' )
    #f.write(result)
    #f.close()
    print(result)#文件输出将其注释

```