《算法设计与分析》第1次作业参考答案

算法分析题

题目1：以下算法的时间复杂度是:

Algorithm 1 : Test(n)

Input: 上界 n

Output: 计算结果 x

  $x \leftarrow 2$

  while $x < \left\lfloor \frac{n}{2} \right\rfloor$ **do**

    $x = 2 * x$

  end while

  return x

答：$\mathrm{O}(\log_2 n)$

解析：在程序中，执行频率最高的语句为 $x = 2 * x$。设该语句共执行了t次，则$2^{t+1} = n/2$，故$t = \log_2(n/2) - 1 = \log_2 n - 2$，得 $T(n) = O(\log_2 n)$。

题目2：证明：$n! = O(n^n)$

证明过程：

Stirling Approximation：

$n! = \sqrt{2\pi n}(\frac{n}{e})^n[1 + \theta(\frac{1}{n})])$

$\lim_{n\to\infty} n!/n^n = \frac{\sqrt{2\pi n}(\frac{n}{e})^n[1+\theta(\frac{1}{n})])}{e^n} = 0$

题目3：对于下列各组函数$f(n)$和$g(n)$，确定$f(n) = O(g(n))$或$f(n) = \Omega(g(n))$或$f(n) = \Theta(g(n))$，并简述理由。

(1) $f(n) = \log n^2$; $g(n) = \sqrt{n}$

(2) $f(n) = n$; $g(n) = \log^2 n$

(3) $f(n) = 10$; $g(n) = \log 10$

(4) $f(n) = 2^n$; $g(n) = 3^n$

答：

(1)$\log n^2 = O(\sqrt{n})$

(2)$n = \Omega(\log^2 n)$

$(3) 10 = \Theta(\log 10)$

$(4) 2^n = O(3^n)$

题目4：一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。例如，第 6 页用数字 6 表示，而不是 06 或者 006 等。数字计数问题要求对给定书的总页码 n，计算出书的全部页码中分别用到多少次数字 $0, 1, 2, ..., 9$。

参考答案：

算法思路：

- 暴力: 用数组$A[0:9]$存储数字与出现次数的对应关系，$A[0]$表示0出现的次数；从1到总页码数n遍历每个页码，对每个页码中出现的数字的次数进行统计，存入数组A。此方法效率较低。

- 递归法：由$0, 1, 2, ...9$组成的所有n位数。从n个0到n个9共有10^n个n位数。在这10^n个n位数中，$0, 1, 2, ...9$每个数字使用次数相同，设为$f(n)$。$f(n)$满足如下递归式：

  $$f(n) = \begin{cases} 10f(n-1) + 10^{n-1} & n > 1 \\ 1 & n = 1 \end{cases}$$

  由此可知，$f(n) = n10^{n-1}$ 据此，可从高位向低位进行统计，再减去多余的0的个数即可。 (可参考此blog)

伪代码示例：

Algorithm 2 暴力法

Input: 总页码数 n

Output: $0, 1, 2, ..9$ 每个数字出现的次数

```
A ← int[10];                    //建立数字与出现次数的对应关系
for i = 1 → n do
    while i ≠ 0 do
        A[i%10] + +
        i ← i/10
    end while
end for
return A
```

Algorithm 3 递归法

Input: 总页码数 n

Output: $0, 1, 2, ..9$ 每个数字出现的次数

function SOLVE(n, A[])
    $len \leftarrow \log_{10} n + 1$
    $highNum \leftarrow n/10^{len-1}$
    for $i = 0 \rightarrow 9$ **do**
        $A[i] \leftarrow A[i] + highNum * (len - 1) * 10^{len-2}$
    end for
    for $i = 0 \rightarrow hignNum$ **do**
        $A[i] \leftarrow A[i] + 10^{len-1}$
    end for
    $remain \leftarrow n\%10^{len-1}$
    if $t == 0$ **then**
        $A[highNum] + +$
        $A[0] \leftarrow A[0] + len - 1$
        return
    end if
    $lenOfRemain \leftarrow \log_{10} t + 1$
    if $lenOfRemain \neq (len - 1)$ **then**
        $A[0] \leftarrow A[0] + (len - lenOfRemain - 1) * (remain + 1)$
        $A[highNum] \leftarrow A[highNum] + remain + 1$
        return SOLVE(remain)
    end if
end function
function DECREASEZERO(A[], len)
    for $i = 0 \rightarrow len$ **do**
        $A[0] \leftarrow A[0] - 10^i$
    end for
end function