算法课程设计

周翔辉

11603080122

2018 年 12 月

目录

1 基本的递归算法

1.1 二项式的计算

完成二项式公式计算, 即 $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$ 公式解释为了从 n 个不同元素中抓取 k 个元素 (C_n^k), 可以这样考虑, 如果第一个元素一定在结果中, 那么就需要从剩下的 $n-1$ 个元素中抓取 $k-1$ 个元素 (C_{n-1}^{k-1}); 如果第一个元素不在结果中, 就需要从剩下的 $n-1$ 个元素中抓取 k 个元素 (C_{n-1}^k)。要求分别采用以下方法计算, 并进行三种方法所需时间的经验分析。

1.1.1 直接采用递归算法

分析题目可知，计算二项式公式的递推公式为 $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$ ，而递归的终止条件就为当 $k = 1$ and $k < n$ 或 $k = n$ 或 $n - k = 1$ 时分别返回 $n, 1, n$ 。

算法 CalculateBinomialRecursion (k, n)

// 计算二项式公式使用递归
// 输入：二项式公式的 k, n
// 输出：二项式公式的值
if $k > 0$ && $n > 0$
    if $k = 1$ and $n = 1$
        return 1
    else if $k = 1$
        return n
    else
        return CalculateBinomialRecursion $(k-1, n-1)$
        + CalculateBinomialRecursion $(k, n-1)$
return 0

```go
func CalculateBinomialRecursion(k, n int) int {
    if k > 0 && n > 0 {
        if k == 1 && n == 1 {
            return 1
        } else if k == 1 {
            return n
        } else {
            return CalculateBinomialRecursion(k-1, n-1)
            + CalculateBinomialRecursion(k, n-1)
        }
    }
```

```
12          return 0
13  }
```

1.1.2 采用备忘录方法

1.1.3 采用迭代算法

1. 直接采用递归算法
2. 采用备忘录方法
3. 采用迭代算法