算法 读书笔记

目录

前言

作为教材

- 这些算法一般都巧妙奇特，20 行左右的代码就足以表达。
- 学过一门计算机方面的先导课程就足矣，只要熟悉一门现代编程语言并熟知现代计算机系统，就都能够阅读本书。
- 本书涉及的内容是任何准备主修计算机科学、……、等专业的学生应了解的基础知识，……

背景介绍

- ……而本书则是专门为大学一、二年级学生设计的一学期教材，也是最新的基础入门书或从业者的参考书。

第一章 基础

- 算法：有限、确定、有效的并适合用计算机实现的解决问题的方法
- 数据结构：便于算法操作的组织数据的方法
- 数据结构是算法的副产品或是结果，简单的算法也会产生复杂的数据结构，复杂的算法也许只需要简单的数据结构。
- 算法分析：为一项任务选择最合适的算法而进行的分析
- 学习算法是非常有趣和令人激动的，因为这是一个历久弥新的领域（我们学习的绝大多数算法都还不到“五十岁”……）

1.1.4 简便记法

- 程序有很多写法，我们追求清晰、优雅和高效的代码

1.1.5.1 创建并初始化数组

- 在代码中使用数组时，一定要依次声明、创建并初始化数组。忽略了其中的任何一步都是很常见的编程错误。

1.1.6.1 静态方法

- 方法需要**参数**（某种数据类型的值）并根据参数计算出某种数据类型的**返回值**（例如数学函数的结果）或者产生某种**副作用**（例如打印一个值）
- 典型静态方法的实现
  - 判断一个数是否是素数

```
public static boolean isPrime(int N)
{
  if (N < 2) return false;
  for (int i = 2; i*i <= N; i++)
    if (N % i == 0) return false;
  return true;
}
```

  - 计算平方根（牛顿迭代法）

```
public static double sqrt(double c)
{
  if (c < 0) return Double.NaN;
  double err = 1e-15;
  double t = c;
  while (Math.abs(t - c/t) > err * t)
    t = (c/t + t) / 2.0;
  return t;
}
```

  - 牛顿法（Newton's method） 求 a 的 m 次方根

$$x_{n+1} = x_n - \frac{x_n}{m}(1 - a{x_n}^{-m})$$

当 $m = 2$ 时，则：

$$\begin{aligned} x_{n+1} &= x_n - \frac{x_n}{2}(1 - a{x_n}^{-2}) \\ &= \frac{x_n}{2} + \frac{a{x_n}^{-1}}{2} \\ &= (a/x_n + x_n)/2 \end{aligned}$$

1.1.6.4 递归

- 递归代码比相应的非递归代码更加简洁优雅、易懂
- 编写递归代码时最重要的有以下三点
  - 递归总有一个**最简单的情况**——方法的第一条语句总是一个包含 return 的条件语句
  - 递归调用总是尝试解决一个**规模更小**的子问题，这样递归才能收敛到最简单的情况
  - 递归调用的父问题和尝试解决的子问题之间不应该有**交集**
- 坚持这些原则能写出清晰、正确且容易评估性能的程序
- 使用递归的另一个原因：可以使用数学模型来估计程序的性能
- **二项分布**。估计用以下代码计算 binomial(100, 50, 0.25) 将会产生的递归调用次数：

```
public static double binomial(int N, int k, double p)
{
  if (N == 0 && k == 0) return 1.0;
  if (N < 0 || k < 0) return 0.0;
  return (1.0 - p)*binomial(N-1, k, p) + p*binomial(N-1, k-1, p);
}
```

  将已经计算过的值保存在数组中并给出一个更好的实现。

- 解：将 $binomial(N,k,p)$ 产生的递归调用次数记为 $c(n,k)$，当 $k>0$ 时，则：

$$\begin{aligned}c(n,k)&=1+c(n-1,k)+c(n-1,k-1)\\&=2+c(n-2,k)+c(n-2,k-1)+c(n-1,k-1)\\&=...\\&=n+c(n-n,k)+c(n-n,k-1)+...+c(n-1,k-1)\\&=n+1+c(-1,k)+c(-1,k-1)+c(0,k-1)+...+c(n-1,k-1)\\&=n+3+\sum_{i=0}^{n-1}c(i,k-1)\end{aligned}$$

  - 二项分布 设 X 为 n 重贝努利试验中成功的次数，则称 X 取值的概率分布为**二项分布**，记为 $b(n,p)$。
    - n 重贝努利试验的特点
      1. 重复进行 n 次相互独立的试验
      2. 每次试验只有成功或失败中的一个结果
      3. 每次出现成功的概率均为 p
  - $binomial(n,k,p)$ 表示事件 “n 重贝努利试验中成功出现 k 次” 的概率，则： 对于其中任意一次贝努利试验，成功概率为 p， 若其成功，则事件变为 $n-1$ 重贝努利试验中成功出现 $k-1$ 次； 若其失败，则事件变为 $n-1$ 重贝努利试验中成功出现 k 次。 故：

$$binomial(n,k,p)=p*binomial(n-1,k-1,p)+(1-p)*binomial(n-1,k,p)$$

  又 $n\geqslant k$，则需判断： $k<0$ 时，概率为 0 $n<k$ 时，概率为 0 $n=0,k=0$ 时，概率为 1