数据结构与算法

Data Structure and Algorithm

IV. 递归

授课人：Kevin Feng

翻译　：梁 少 华

Review

回顾

- 数据结构与算法
- 数学回顾
  - 大O表示法
  - 时间复杂度
  - 空间复杂度
- 数组
- 数组列表
- 搜索和排列

递归 VS. 迭代

- 递归函数是一种自我调用的函数
- 阶乘
  - 数学定义
    - 0! = 1
    - N! = N (N - 1)! if N > 0
- 斐波那契
  - f(0) = 0, f(1) = 1
  - f(n) = f(n-1) + f(n-2), for n>1

递归的基本思想

- 可以把要解决的问题转化为一个子问题，而这个子问题的解决方法仍与原来的解决方法相同，只是问题的规模变小了。
- 原问题可以通过子问题解决而组合解决。
- 基础案例：存在一种简单的情境，是问题在简单情境下退出。

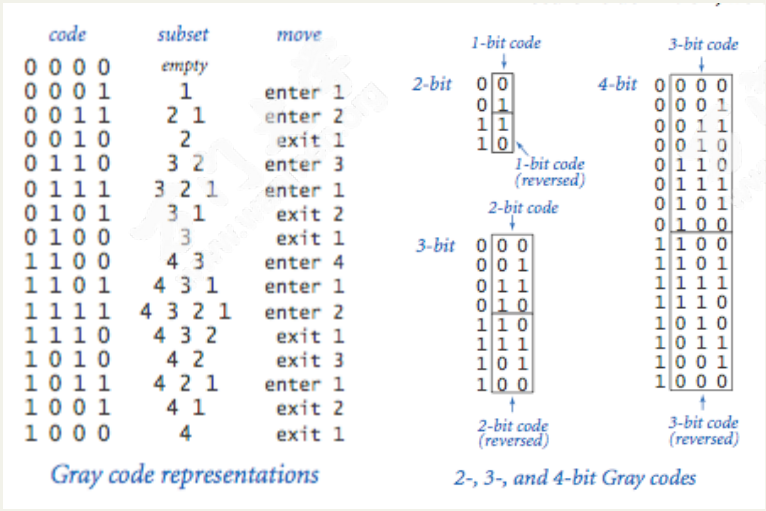
- 如果你对程序逻辑不够小心,你可能会错过一个基本的例子并进入无限递归
- 编码时要非常小心,因为它很难调试

递归应用

- 标尺细分

```
1
1 2 1
1 2 1 3 1 2 1
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

- 整数序列转换
  - 给定两个整数 a ≤ b, 编写一个程序,通过加1和乘以2的方式，将a变换成b.
  - 例如,　　23 = ((5 * 2 + 1) * 2 + 1)
    113 = ((((11 + 1) + 1) + 1) * 2 * 2 * 2 + 1)
- 汉诺塔
- 格雷码: 两个相邻数的代码只有一位二进制数不同。



code	subset	move
0 0 0 0	empty	
0 0 0 1	1	enter 1
0 0 1 1	2 1	enter 2
0 0 1 0	2	exit 1
0 1 1 0	3 2	enter 3
0 1 1 1	3 2 1	enter 1
0 1 0 1	3 1	exit 2
0 1 0 0	3	exit 1
1 1 0 0	4 3	enter 4
1 1 0 1	4 3 1	enter 1
1 1 1 1	4 3 2 1	enter 2
1 1 1 0	4 3 2	exit 1
1 0 1 0	4 2	exit 3
1 0 1 1	4 2 1	enter 1
1 0 0 1	4 1	exit 2
1 0 0 0	4	exit 1

Gray code representations

2-, 3-, and 4-bit Gray codes

排列与组合

- 子集
  - 给定一组不同的整数, 返回所有可能的子集.
- 子集 II
  - 给定一个可能包含重复数字的整数集合,返回所有可能的子集.
- 排列
  - 给定 abc
  - 输出: bca cba cab acb bac abc
- 大小为k的排列
  - 取两个参数n和k, 打印出所有的P(n, k) = n! / (n-k)! 种不同的排列组合。
  - 当 k = 2 and n = 4
  - ab ac ad ba bc bd ca cb cd da db dc

递归应用 II

- 字符串中字母的大小写组合
  - 枚举输入中指定的任何字母的全部大写/小写排列
  - 例如,
  - word = “medium-one”
  - Rule = “io”
  - solutions = [“medium-one”, “medIum-one”, “medium-One”, “medIum-One”]
- 括号
  - 给定一个数字n,要求写一个方法,生成所有n个括号的可能合法组合.

递归 VS. 迭代

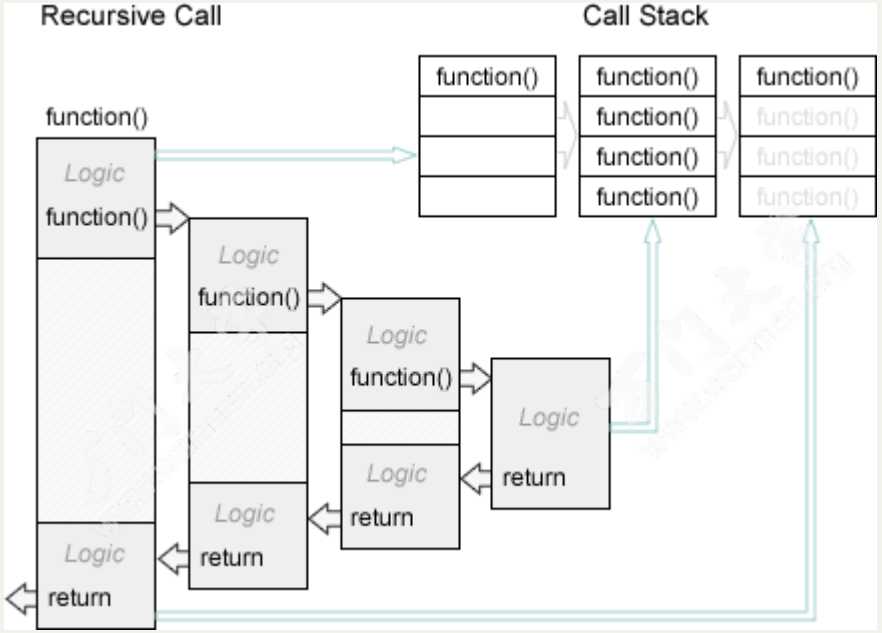
- 递归和迭代执行相同任务
  - 一次解决一个复杂的任务,并将结果结合起来.
- 迭代的重点
  - 不断重复,直到任务“完成”为止
  - 例如,循环计数器达到极限,链接列表达到空指针,instream.eof() 变为真
- 递归的重点
  - 解决一个大问题,把它分解成更小更小的部分,直到你能解决它;结合这些结果.
  - 例如,递归阶乘函数.

哪一个更好

- 没有明确答案
- “数学家”通常更喜欢递归方法.
  - 解决方案往往较短, 更接近抽象数学实体.
  - 直观
  - 很难测试/调试
- “程序员”, 特别是有过丰富编程经验的, 通常更喜欢迭代解决方案
  - 不知何故,它似乎更吸引许多人
  - 本地循环, 不那么 “神奇”
- 我应该选择哪种方法

我应该选择哪种方法

- 取决于问题
- 如果用递归地实现方法,许多ADT（例如树）更简单和更自然.
- 注意: 递归通常要慢得多.



作业 I

格雷码

- 打印格雷码 (不仅仅是改变的比特位置的序列).

海明距离

- 长度为n的两个字符串之间的海明距离等于两个字符串对应位置的不同字符的个数.写一个程序,从命令行读入一个整体K和一个字符串S,并从S中打印出最多有K个海明距离的所有字符串.例如,如果K是2并且S是0000那么你的程序应该打印出来:

  0011 0101 0110 1001 1010 1100

- 提示: 选择S中N位的K来翻转.

冯诺依曼序数

- 冯诺依曼整数定义如下: 对于 i = 0,它是空集; 对于 i > 0, 它是包含冯诺依曼整数 0 到 i-1的集合.

- 0 = {} (空集)
- 1 = {0} = { {} }
- 2 = {0,1} = { {}, { {} } }
- 3 = {0,1,2} = {{}, { {} }, { {}, { {} } }}
- 4 = {0,1,2,3} = { {} , { {} }, { {}, { {} } } , {{}, { {} }, { {}, { {} } }} }

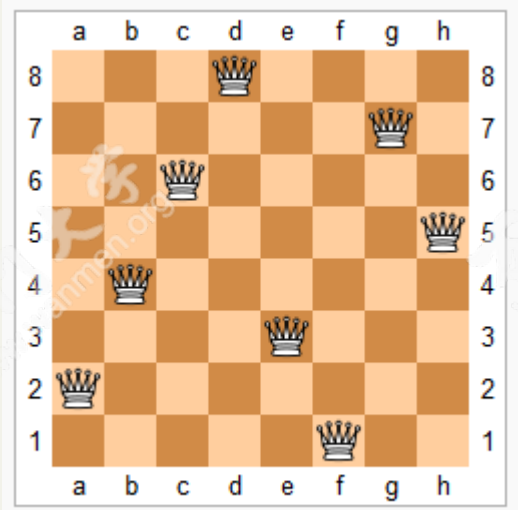
作业 II

- 美分的组成
  - 给定无限数量的25美分，10美分，5美分和1美分,有多少种方法能给出n美分
- 字典序排列
  - 编写一个采用命令行参数N并打印出所有N的程序! 整数0 到 N-1 按字典顺序排列.

012 021 102 120 201 210

作业 III

- N 皇后 (救命，好难啊！)
  - 在一个N*N的棋盘上放置N个皇后，每行一个并使其不能互相攻击（同一行、同一列、同一斜线上的皇后都会自动攻击）。
  - 给定一个整数n, 表示一个n×n的棋盘，问有多少种摆放方案使得棋盘上的任意皇后不能攻击其他皇后。



One solution to the eight queens puzzle

概述

- 递归
- 子集
- 排列
- 组合

数据结构与算法

Data Structure and Algorithm

IV. 递归结束

授课人：Kevin Feng

翻译　：梁少华