求组合问题的不同算法比较分析*

张世良
(宁德师范高等专科学校,福建 宁德 352100)

摘 要:介绍了递归法与回溯法的一般思想,分析了用递归法与回溯法求解组合问题,还对求解问题的复杂度以及优缺点进行了分析比较。

关键词:递归法;回溯法;组合问题;算法比较分析
中图分类号:TP301 **文献标识码**:B

Comparison and analysis of different algorithm for seeking the combination problem

ZHANG Shi Liang
(Ningde Normal College, Ningde 352100,China)

Abstract: It mainly introduces the general thoughts of reduce arithmetic and backtrack algorithm in this article, and solves the combination question by comparing and analyzing the two thoughts, as well as compares them to find the complexity and pros and cons in solving problem.

Key words: reduce arithmetic; backtrack algorithm; combination question; algorithm comparison and analysis

一般来说,无论根据理论的观点还是实践的观点,组合问题都是计算领域中的难题。这是由于通常情况下,随着问题规模的增大,组合对象的数量增长极快,即使是中等大小的实例,其组合的规模也会达到不可思议的数量级。另外还没有一种已知算法能在可接受的时间内,精确地解决绝大多数这类问题。但有些组合问题能用效率较高的算法解决。本文就对回溯法与递归法解决组合问题进行了比较分析[1]。

问题描述:找出从自然数 1,2,…,m 中任取 k 个数的所有组合。

1 用纯递归法求解

能采用递归描述的算法通常拥有的特征为:为求解规模为 N 的问题,设法将它分解成规模较小的问题,然后从这些小问题的解中方便地构造出大问题的解,并且这些规模较小的问题也能采用同样的分解和综合方法,分解成规模更小的问题,并从这些更小问题的解构造出规模较大问题的解。特别地,当规模 N=1 时,能直接得到解。

设函数为 void comb(int m,int k)为找出从自然数 1,2,…,m 中任取 k 个数的所有组合。当组合的第 1 个数字选定时,其后的数字是从余下的 m-1 个数中取 k-1 个数的组合。这就将求 m 个数中取 k 个数的组合问题转化成求 m-1 个数中取 k-1 个数的组合问题。设函数引入工作数组 a[]存放求出的组合的数字,约定函数将确定的 k 个数字组合的第 1 个数字放在 $a[k]$中,当一个组合求出后,才将 a[]中的一个组合输出。第 1 个数可以是 m,m-1,…,k,函数将确定组合的第 1 个数字放入数组后,有两种可能的选择,因还未去掉组合的其余元素,继续递归去确定;或因已确定了组合的全部元素,输出这个组合。细节见以下程序中的 comb 函数。

```
public static void comb(int m, int k)
 {
    int i = 0, j = 0;
    for (i = m; i >= k; i--)
    {
       a[k] = i;
       if (k > 1)
       {
          comb(i - 1, k - 1);
       }
       else
```

* 基金项目:宁德师范高等专科学校科研资助项目(2008Y009)

《电子技术应用》 www.ChinaAET.com


```
        {
            for (j = a[0]; j > 0; j--)
            {
                Console.Write(a[j].ToString() + " ");
            }
            Console.WriteLine("\n\r");
        }
    }
```

2 用回溯法求解

回溯法也称为试探法，该方法首先暂时放弃关于问题规模大小的限制，并将问题的候选解按某种顺序逐一枚举和检验。当发现当前候选解是解的可能性不存在时，就选择下一个候选解；倘若当前候选解除了不满足问题规模要求外，满足其他所有要求时，继续扩大当前候选解的规模，并继续试探。如果当前候选解满足包括问题规模在内的所有要求时，该候选解就是问题的一个解。在回溯法中，放弃当前候选解，寻找下一个候选解的过程称为回溯。扩大当前候选解的规模，以继续试探的过程称为向前试探。

(1)回溯法的方法

对于具有完备约束集 D 的一般问题 P 及其相应的状态空间树 T，利用 T 的层次结构和 D 的完备性，在 T 中搜索问题 P 的所有解的回溯法可以描述为：从 T 的根出发，按深度优先的策略，系统地搜索以其为根的子树中可能包含着回答结点的所有状态结点，而跳过对肯定不含回答结点的所有子树的搜索，以提高搜索效率。具体地说，当搜索按深度优先策略到达一个满足 D 中所有有关约束的状态结点时，即“激活”该状态结点，以便继续往深层搜索；否则跳过对以该状态结点为根的子树的搜索，而一边逐层地向该状态结点的祖先结点回溯，一边“杀死”其子结点已被搜索遍的祖先结点，直到遇到其子结点未被搜索遍的祖先结点，即转向其未被搜索的一个子结点继续搜索。

在搜索过程中，只要所激活的状态结点又满足终结条件，那么它就是回答结点，应该把它输出或保存。由于在回溯法求解问题时，一般要求出问题的所有解，因此在得到回答结点后，同时也要进行回溯，以便得到问题的其他解，直至回溯到 T 的根且根的所有子结点均已被搜索过为止。

(2)回溯法的一般流程和技术

在用回溯法求解有关问题的过程中，一般是一边建树，一边遍历该树。在回溯法中一般采用非递归方法。回溯法的非递归算法的一般流程为：

在用回溯法求解问题，也即在遍历状态空间树的过程中，如果采用非递归方法，则一般要用到栈的数据结构。这时，不仅可以用栈来表示正在遍历的树的结点，而且可以很方便地表示建立该子结点和回溯过程。

若采用回溯法找问题的解，将找到的组合从小到大存于 $a[0],a[1],\cdots,a[r-1]$ 中，组合的元素满足以下性质：

(1)$a[i+1]>a[i]$，后一个数字比前一个大；

(2) $a[i]-i\leqslant n-r+1$。

按回溯法的思想，找解过程可以叙述如下：首先放弃组合数个数为 r 的条件，候选组合从只有一个数字 1 开始。因该候选解满足除问题规模之外的全部条件，扩大其规模，并使其满足上述条件(1)，候选组合改为 1,2。继续这一过程，得到候选组合 1,2,3。该候选解满足包括问题规模在内的全部条件，因而是一个解。在该解的基础上，选下一个候选解，因 $a[2]$ 上的 3 调整为 4，以及以后调整为 5 都满足问题的全部要求，得到解 1,2,4 和 1,2,5。由于对 5 不能再作调整，就要从 $a[2]$ 回溯到 $a[1]$，这时，$a[1]=2$，可以调整为 3，并向前试探，得到解 1,3,4。重复上述向前试探和向后回溯，直至要从 $a[0]$ 再回溯时，说明已经找完问题的全部解。按上述思想写成程序如下：

```
public static  void comb(int n, int r)
 {
    int i=0, j=0;
    for (j = 0; j < r; j++)
      c[j] = j+1;
    for (j = 0; j < r; j++)
    {
      Console.Write(c[j].ToString() + " ");
    }
      Console.WriteLine("\n\r");
      i = r - 1;
    do
    {
      if (c[i]-i < n - r+1)
      {
          c[i]++;
          for (j = i + 1; j < r; j++)
              c[j] = c[j - 1] + 1;
          for (j = 0; j < r; j++)
          {
            Console.Write(c[j] + " ");
          }
          Console.WriteLine("\n\r");
          i = r - 1;
      }
      else --i;
    } while (i >= 0);
 }
```

运行环境:本机 Microsoft Windwos XP 版本 2002 Service Pack 2，CPU 2.13GHz,760M RAM。运行结果归纳后如表 1 所示。

从运行的情况来看，当选出数字个数较少时递归算法相对较快些，当选出数据量相对较大时用回溯法相对快些。而使用递归法，代码更加简洁、明了。

（下转第 56 页）

《电子技术应用》 www.ChinaAET.com

（上接第 52 页）

表 1 两种等法运算速度对比

《电子技术应用》 www.ChinaAET.com

用时/s 使用方法	在 10 个数中取 5 个	在 20 个数中取 5 个	在 30 个数中取 5 个	在 40 个数中取 5 个	在 1 000 个数中取 2 个	在 100 个数中取 2 个	在 20 个数中取 8 个	在 30 个数中取 25 个
用递归算法	4.32	107.39	960.61	3 529.49	1 270.47	190.73	1 232.32	912.41
用回溯算法	4.37	102.70	955.47	3 094.56	1 856.73	280.96	1 157.57	898.77

递归方法中递归调用的空间复杂度是 $O(k-m)$ 的线性阶,因此其时间复杂度为 $O(\log m)$,而回溯算法的空间复杂度为 $O(m^2)$,它的时间复杂度为 $O(m\times k)$[2-3]。

递归方法适用于问题的规模缩小到一定的程度就可以容易地解决的问题，该问题可以分解为若干个规模较小的相同问题,即该问题具有最优子结构性质。利用该问题分解出的子问题的解可以合并为该问题的解，该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题。递归法的关键是必须有一个递归终止条件,即要有递归出口,无条件的递归是毫无意义的。递归方法设计算法的策略不仅适用于计算数学问题，而且也适用于非数值运算领域。

递归法的优点是结构清晰,可读性强,而且容易用数学归纳法来证明算法的正确性。其缺点是递归算法的运行效率较低，无论是耗费的计算时间还是占用的存储空间都比非递归算法多。

回溯法的优点是适用于不可能使用实验研究法的许多情形。其缺点是缺乏对自变项的控制,难确定有关的因素、事件的发生[4],因此要根据解决问题自身的特点选用合适的算法。

参考文献

[1] 朱战立.数据结构(C++语言描述) [M].北京:高等教育出版社,2004.

[2] ANANY L. Introduction to the design and analysis of algorithms[M] . 北京:清华大学出版社,2005.

[3] 朱玉龙,任文岚. 递归程序设计的公式化方法[J] .小型微型计算机系统,2001 ,22 (11) :1 389-1 390.

[4] ALSUWAIYEL M.H. Algorithms design techniques and analysis [M].吴伟旭,方世昌,译.北京:电子工业出版社,2004.

(收稿日期:2009-02-22)

《电子技术应用》 www.ChinaAET.com