中山大学学报 (自然科学版)
第 35卷 第 6期 ACTA SCIENTIARUM NATURALIUM Vol. 35 No. 6
1996年 11月 UNIVERSITATIS SUNYATSENI Nov. 1996

构造二叉树的一个算法*

娄 定 俊
(中山大学计算机科学系,广州 510275)

摘 要 给出一个算法,该算法输入一棵二叉树的前序遍历和中序遍历的结点序列,构造出该二叉树,该算法具有 $O(n)$时间复杂度,是解决该问题的最优算法,其中 n为二叉树的结点数.

关键词 前序遍历,中序遍历,二叉树.
分类号 TP 301.6

在数据结构的教学中,常常遇到这样的问题:给出一棵二叉树的前序遍历和中序遍历的结点序列,用人工构造出这棵二叉树[1]. Knuth[2]证明了:给出一棵二叉树的前序遍历和中序遍历的结点序列,可以构造出这棵二叉树.显而易见,存在一个算法,输入一棵二叉树的前序遍历和中序遍历的结点序列,构造出这棵二叉树.但通常的算法需要 $O(n^2)$时间,其中 n为二叉树的结点数.本文中我们给出这样一个算法,该算法通过巧妙地控制输入的结点序列和利用栈,使该算法的时间复杂度为 $O(n)$.可以证明该算法是解决以上问题的最优算法.以下用 Pascal语言给出该算法.

```
PROGRAM ConsBiTree;
CONST
  max= 999; /* 假设二叉树的结点数 n不大于 max* /
TYPE
  link= ^ node;
  node= RECORD    /* 二叉树结点的类型 * /
          value: char;
          lchild: link;
          rchild: link;
        END;
VAR
  Fod, Pod: ARRAY [1..max] OF char;
  S: ARRAY [1..max] OF link;    /* S是栈* /
  i, j, n, top: integer;
  P, Q, root: link;
```

* 收稿日期: 1995-12-27 娄定俊,男,34岁,副教授

```
BEGIN
  new(P); P^.value:='$'; {*"$"是不出现在二叉树中的符号*}
  top:=1; S[top]:=P; {*栈底放"$"*}
  READ(n); {*n为二叉树的结点数*}
  FOR i:=1 TO n DO READ(Fod[i]);
  {*前序遍历的结点序列放在Fod[1],Fod[2],…,Fod[n]中*}
FOR j:=1 TO n DO READ(Pod[j]);
  {*中序遍历的结点序列放在Pod[1],Pod[2],…,Pod[n]中*}
  i:=1; j:=1;
new(P); P^.value:=Fod[i];
  root:=P; {*root指向树根*}
  WHILE (i<=n) OR (j<=n) DO
  BEGIN
    top:=top+1; S[top]:=P; {*Fod[i]进栈*}
    WHILE Fod[i]<>Pod[j] DO
    BEGIN
      i:=i+1;
       new(P); P^.value:=Fod[i];
      S[top]^.lchild:=P; {*Fod[i-1]的左孩子为Fod[i]*}
      top:=top+1; S[top]:=P; {*Fod[i]进栈*}
    END;
    S[top]^.lchild:=NIL; {*Fod[i]的左孩子为空*}
    i:=i+1; j:=j+1;
    Q:=S[top]; top:=top-1; {*弹出栈顶元素*}
    WHILE (j<=n) AND (Pod[j]=S[top]^.value) DO
    BEGIN
      Q^.rchild:=NIL; {*Q的右孩子为空*}
      Q:=S[top]; top:=top-1; {*弹出栈顶元素*}
      j:=j+1;
    END;
    IF (i<=n) OR (j<=n) THEN
    BEGIN
       new(P); P^.value:=Fod[i];
       Q^.rchild:=P; {*Q的右孩子为Fod[i]*}
    END
    ELSE Q^.rchild:=NIL;
  END;
END.
```

在以上算法中不考虑输入语句,内层第一个 WHILE循环每循环一次,i的值增加 1.内层的第二个 WHILE循环每循环一次,j的值增加 1.外层的 WHILE循环和内层的两个 WHILE循环的循环次数之和为 $2n$.因而算法的时间复杂度为 $2n$,是 $O(n)$数量级的.注意到每个解决该问题的算法都必须读前序遍历和中序遍历的结点序列至少各一遍,才能构造出二叉树,因而时间复杂度至少为 $2n$,所以本算法是最优的.

参 考 文 献

1　袁蒲佳,龙玉国,杨薇薇.数据结构.武汉: 华中理工大学出版社, 1991. 101

2　D. E.克努特.计算机程序设计技巧(第一卷　基本算法).管纪文,苏运霖译.北京: 国防工业出版社, 1980. 277

An Algorithm for Constructing Binary Trees

Lou Dingjun[*]

Abstract　An algorithm is given, which inputs the node sequences of a binary tree for preorder traversal and postorder traversal and constructs the binary tree. The algorithm is optimal for the problem with time complexity $O(n)$, where n is the number of the nodes of the tree.

Keywords　preorder traversal, postorder traversal, binary tree

[*] Department of Computer Science, Zhongshan University, Guangzhou 510275