第四章 实现集合的高级方法

§4.1 二叉查找树

\*假设集合有线性序，用二叉查找树表示集合，INSERT, DELETE, MEMBER和MIN操作平均时间复杂度为。

一．二叉查找树

1. 定义：一个二叉查找树是一棵二叉树，用集合元素标记每个结点。每个结点的左子树上的元素都小于该结点上的元素，每个结点右子树上的元素都大于该结点上的元素。

2. 例子：

图4.1

二．二叉查找树的操作

1. 数据结构

type

nodetype=RECORD

element : elementtype;

leftchild : nodetype;

rightchild : nodetype;

END;

SET=nodetype;

2. 操作

FUNCTION MEMBER (x : elementtype; A : SET) : Boolean;

/\*returns true if x is in A, false otherwise\*/

BEGIN

IF A = nil THEN

RETURN(false) /\*x is never in \*/

ELSE IF x = A.element THEN

RETURN(true)

ELSE IF x < A.element THEN

RETURN( MEMBER(x, Aleftchild) )

ELSE /\* x > A.element \*/

RETURN( MEMBER(x, Arightchild) )

END;

PROCEDURE INSERT (x : elementtype; VAR A : SET);

/\* add x to set A \*/

BEGIN

IF A = nil THEN

BEGIN

new(A);

A.element := x;

A.leftchild := nil;

A.rightchild := nil;

END

ELSE IF x < A.element THEN

INSERT(x, A.leftchild)

ELSE IF x > A.element THEN

INSERT(x, A.rightchild);

/\* if x = A.element, we do nothing; x is already in the set \*/

END;

FUNCTION DELETEMIN (VAR A : SET) : elementtype;

/\* returns and removes the smallest element from set A \*/

BEGIN

IF A.leftchild = nil THEN

BEGIN /\* A points to the smallest element from set A \*/

DELETEMIN := A.element;

A := A.rightchild;

/\* replace the node pointed to by A by its right child \*/

END

ELSE /\* the node pointed by A has a leftchild \*/

DELETEMIN := DELETEMIN(A.leftchild)

END;

PROCEDURE DELETE (x : elementtype; VAR A : SET);

/\* removes x from set A \*/

BEGIN

IF A <> nil THEN

IF x < A.element THEN

DELETE(x, A.leftchild)

ELSE IF x > A.element THEN

DELETE(x, A.rightchild)

/\* if we reach here, x is at the node pointed by A \*/

ELSE IF (A.leftchild = nil) AND (A.rightchild = nil) THEN

A := nil /\* delete the leaf holding x \*/

ELSE IF A.leftchild = nil THEN

A := A.rightchild

ELSE IF A.rightchild = nil THEN

A := A.leftchild

ELSE /\* both children are present \*/

A.element := DELETEMIN(A.rightchild)

END;

三．二叉查找树操作的时间复杂性

1. 最坏情况

当输入元素是有序的(从小到大)，则二叉查找树退化为一个链表，各种操作的时间复杂度同链表表示法。

例入：MEMBER操作需O(n)时间，插入第个元素需时间，插入n个元素需

时间。

2. 平均时间复杂性

若二叉树是完全树，从根结点到叶子的路径上结点数为，故MEMBER, INSERT, DELETE, DELETEMIN只需时间。

下面证明，在随机的情况下建立二叉查找树，从根结点到任一结点的路径上平均结点数为。假设：所建立的二叉查找树只是由插入操作建立的，插入时n个结点的各种排列顺序的可能性相同。令P(n)为n个结点的二叉查找树从根结点到任一结点的路径的平均结点数。则。

设是输入二叉查找树的第一个结点，有个结点小于，有个结点大于。

由归纳假设，的左子树从根到任一结点路径平均结点数为，的右子树从根到任一结点路径平均结点数为。

当左子树有个结点，右子树有个结点时，从根到任一结点路径的结点数为

下面归纳证明： 。

当时，。

对任意，设 。

故

当时，

当时，

故MEMBER, INSERT, DELETE, DELETEMIN, MIN操作平均时间复杂度均为。