PES UNIVERSITY

Department of Computer Science and Engineering

PES UNIVERSITY

UE19CS202: Data Structures and its Applications (4-0-0-4-4)

Trees
BST: Implementation using Arrays

Dr. Shylaja S S
Ms. Kusuma K V

Implicit Array Representation of BST

In the implicit array representation, an array element is allocated whether or not it serves to contain a node of a tree. We must, therefore, flag unused array elements as nonexistent, or null, tree nodes. This may be accomplished by adding a logical flag field, used, to each node. Each node then contains two fields: info and used.

```
typedef struct node
{
        int info;
        int used;
}NODE;
```

NODE[p].used is TRUE if node p is not a null node and FALSE if it is a null node.

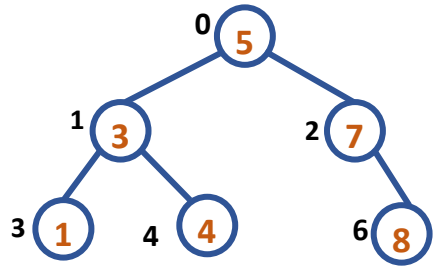

Figure 2: Binary Search Tree

Position: p	0	1	2	3	4	5	6
info	5	3	7	1	4		8
used	1	1	1	1	1	0	1

Array Representation of Binary Search Tree in Figure 2

//BST Array Implementation

```
#include<stdio.h>

typedef struct tree_node
{
        int info;
        int used;
}TREE;

#define MAXNODES 50

void init(TREE t[MAXNODES])
{
        for(int i=0;i<MAXNODES;i++)
                t[i].used=0;
}
```

```
void create(TREE *bst)
{
	int ele, wish;

	printf("Enter the root element\n");
	scanf("%d",&bst[0].info);
	bst[0].used=1;

	do{
		printf("Enter an element\n");
		scanf("%d",&ele);

		int p=0;

		while(p<MAXNODES && bst[p].used)
		{
			if(ele<bst[p].info)
				p=2*p+1;
			else
				p=2*p+2;
		}

		if(p>=MAXNODES)
			printf("Insertion not possible\n");
		else
		{
			bst[p].info=ele;
			bst[p].used=1;
		}
		printf("Do you wish to add another\n");
		scanf("%d",&wish);
	}while(wish);
}
void inorder(TREE* bst, int r)
{
	if(bst[r].used)
	{
		inorder(bst,2*r+1);
		printf("%d ",bst[r].info);
		inorder(bst,2*r+2);
	}
}
```

```
int main()
{
        TREE bst[MAXNODES];

        init(bst);
        create(bst);
        inorder(bst,0);
        return 0;
}
```

Note that under this representation it is always required to check that the range (NUMNODES) has not been exceeded whenever we move down the tree.

Array Implementation is suitable for a complete binary tree. Otherwise there will be many vacant positions in between. In such cases we may look into an alternate representation of tree nodes, i.e., the Linked representation, which makes use of the left and right pointers along with the info field.