



รายงานเรื่อง

Binary Tree: Insert, Delete, Show [Array]

รายชื่อสมาชิก

1.นางสาวณัฐชานันท์ ล้อดี	6530200118
2.นางสาววิรญา เล็กชะอุ่ม	6530200479
3.นายสุภกฤต สมฤทธิ์จินดา	6530200495
4. นางสาวกฤติมา เขาวนั่ดี	6530200568
5.นายณกรณ์ ตั้งประภาพร	6530200614
6.นายพนธ์ปวิธ ศิลาบำรุงราษฎร์	6530200665

เสนอ

ผศ.ดร. จีรพรรณ เจริญสุข

คณะวิทยาศาสตร์ ศรีราชา

มหาวิทยาลัย เกษตรศาสตร์ วิทยาเขตศรีราชา

- Code

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MAX_SIZE 7
```

```
char binaryTree[MAX_SIZE];
```

```
void insert(char value){
```

```
    for(int i=0;i<MAX_SIZE;i++){
```

```
        if(binaryTree[i] == ' '){
```

```
            binaryTree[i] = value;
```

```
            break;
```

```
        }
```

```
        if(i+1 == MAX_SIZE){
```

```
            printf("Binary Tree isn't Empty!\n");
```

```
        }
```

```
    }
```

```
}
```

```
int leftChild(int index) {
```

```
    return 2 * index + 1;
```

```
}
```

```
int rightChild(int index) {
```

```
    return 2 * index + 2;
```

```
}
```

```
void deleteNode(int index) {
```

```
    int leftChildIndex = leftChild(index);
```

```

if (leftChildIndex < MAX_SIZE && binaryTree[leftChildIndex] != ' ') {
    int replacementIndex = leftChildIndex;
    while (rightChild(replacementIndex) < MAX_SIZE && binaryTree[rightChild(replacementIndex)]
!= ' ')
        replacementIndex = rightChild(replacementIndex);
    binaryTree[index] = binaryTree[replacementIndex];

    deleteNode(replacementIndex);
} else if (rightChild(index) < MAX_SIZE && binaryTree[rightChild(index)] != ' ') {
    int replacementIndex = rightChild(index);
    while (leftChild(replacementIndex) < MAX_SIZE && binaryTree[leftChild(replacementIndex)] != '
')
        replacementIndex = leftChild(replacementIndex);

    binaryTree[index] = binaryTree[replacementIndex];
    deleteNode(replacementIndex);
} else {
    binaryTree[index] = ' ';
}
}

```

```

void printTree(char arr[], int n, int index, int space) {
    if (index >= n) {
        return;
    }
    printTree(arr, n, rightChild(index), space + 5);
    for (int i = 0; i < space; i++) {
        printf(" ");
    }
    printf("%c\n", arr[index]);
    printTree(arr, n, leftChild(index), space + 5);
}

```

```

int mode = 1;

char value;

char status;

int main(){

    for(int i=0;i<MAX_SIZE;i++){

        binaryTree[i] = ' ';

    }

    while(mode != 0){

        printf("Binary Tree Program #^$#*&@^\n\n");

        printf(" [1] Insert Value\n");

        printf(" [2] Delete Value\n");

        printf(" [3] Show Binary Tree \n");

        printf(" [0] Exit \n\n");

        B:

        printf("Enter Mode : ");

        int check = scanf(" %d",&mode);

        if(mode < 0 || mode > 3){

            printf("Got Exception !#!^$!^ | Please Try Again!\n");

            goto B;

        }

        if(mode == 1){

            printf("Enter Insert Value (Character/Number(0-9)) : ");

            scanf(" %c",&value);

            insert(value);

            printf("Binary Tree Display \n");

            printTree(binaryTree, sizeof(binaryTree), 0, 0);

        }else if(mode == 2){

            printf("Enter Delete Value (Character/Number(0-9)) : ");

            scanf(" %c",&value);

            for(int i=0;i<MAX_SIZE;i++){

                if(binaryTree[i] == value){

```

```

        deleteNode(i);

        break;
    }if(i+1 == MAX_SIZE){
        printf("Not Found!\n");
    }
}

printf("Binary Tree Display \n");
printTree(binaryTree, sizeof(binaryTree), 0, 0);
}else if(mode == 3){
    printf("Binary Tree Display \n");
    printTree(binaryTree, sizeof(binaryTree), 0, 0);
}else if(mode == 0){
    break;
}
A:
printf("Continue (Y/N) : ");
scanf(" %c",&status);
if(status == 'Y' || status == 'y'){
    continue;
}else if (status == 'N' || status == 'n'){
    break;
}else{
    printf("Got Exception !#!^$!^ | Please Try Again!\n");
    goto A;
}
}
}

```

- Monitor

Main menu

หน้าจอหลักของเมนู โดยมี

- 1.Insert Value
- 2.Deleted Value
- 3.Show Binary Tree
- 4.Exit

```

Binary Tree Program #^$#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 

```

Insert

ทำการเลือก Insert Value Mode โดย ใส่ value = 0,1,2,3,4 ตามลำดับ

โดยจะมีการแสดงผลผ่านทางหน้าจอดังภาพ

```

Enter Mode : 1
Enter Insert Value (Character/Number(0-9)) : 0
Binary Tree Display

0

Continue (Y/N) : y
Binary Tree Program #^$#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 1
Enter Insert Value (Character/Number(0-9)) : 1
Binary Tree Display

0
1

Continue (Y/N) : y
Binary Tree Program #^$#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 1
Enter Insert Value (Character/Number(0-9)) : 2
Binary Tree Display

2
0
1

Continue (Y/N) : 

```

```

Continue (Y/N) : y
Binary Tree Program #^$#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 1
Enter Insert Value (Character/Number(0-9)) : 3
Binary Tree Display

2
0
1
3

Continue (Y/N) : y
Binary Tree Program #^$#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 1
Enter Insert Value (Character/Number(0-9)) : 4
Binary Tree Display

2
0
1
4
3

Continue (Y/N) : 

```

Delete

ทำการเลือก Delete Value Mode โดยทำการลบ value ที่เราได้ทำการ Insert ไว้

โดยเริ่มจากการ ลบ 1 ก่อน -> 3 จะทำการขยับขึ้นมาแทนที่ตำแหน่งของ value 1 จะได้ดังภาพ

ต่อมา ทำการ ลบ 4 ผลลัพธ์จะได้ดังภาพ

```
3
Continue (Y/N) : y
Binary Tree Program #^&#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 2
Enter Delete Value (Character/Number(0-9)) : 1
Binary Tree Display

    2
   /
0 /
  /
   4
  /
   3

Continue (Y/N) : y
Binary Tree Program #^&#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 2
Enter Delete Value (Character/Number(0-9)) : 4
Binary Tree Display

    2      [
   /
0 /
  /
   3

Continue (Y/N) : [
```

Show

ทำการเลือก Show Binary Tree Mode ซึ่งในโหมดนี้จะเป็นการ โชว์ tree หลังจากที่เราทำการ Insert และ Delete เรียบร้อยแล้ว

```
Continue (Y/N) : y
Binary Tree Program #^&#*#&@^

[1] Insert Value
[2] Delete Value
[3] Show Binary Tree
[0] Exit

Enter Mode : 3
Binary Tree Display

    2      [
   /
0 /
  /
   3

Continue (Y/N) : [
```