

# 实验报告#4 - 二叉树

---

[实验报告#4 - 二叉树](#)

[开发环境](#)

[程序说明](#)

[运行逻辑](#)

[数据结构](#)

[算法设计](#)

[运行结果](#)

[作者信息](#)

## 开发环境

---

- 编译器: gcc v6.3.0
- 编辑器: VS Code
- 平台: X86-64
- OS: win10 home

## 程序说明

---

### 运行逻辑

1. 运行程序
2. 输入数据的数量
3. 在一行中输入数据，用空格分开（只接受字母作为输入数据）
4. 程序自动输出建立的树和反向的树（左为根节点）
5. 程序结束

### 数据结构

```

struct Node
{
    char data;
    Node *left, *right;
    Node(char d)    //构造函数
};
class myTree
{
private:
    Node *head;
    static int count;

public:
    myTree();    //构造函数
    void insert(char d);    //插入
    void print();    //打印树
    void PrintNodeByLevel(Node *root, int level);    //打印树的辅助函数
    void reverseTree();    //反转树（递归实现）
    void reverse(Node* root);    //反转树的辅助函数
    void reverseANode(Node * n);    //交换两个节点的函数
};

```

## 算法设计

- 插入函数（树的构建）
  - 判断数据是否合法
  - 判断根节点是否为空
  - 根据数据与当前节点的大小向下遍历
    - 小于则向左节点，大于则向右节点
    - 如果子节点为空，插入
- 反转函数

```

//结构
void reverseTree();    //公有方法，无输入
| - void reverse(Node *root);    //（设计为）私有方法，通过前递归调用自身实现
| - void reverseANode(Node *n);    //（设计为）私有方法，如果一个节点的子节点非空，则交换它们

```

- 打印函数
  - 利用魔改中序遍历实现
  - 增加一个参数来表示当前节点距离根节点的距离（即层数），然后通过这个参数来决定打印空格的数量
  - 树是向左展开的

## 运行结果

---

```
Enter the number of data(only char will be accepted): 6
Enter the data in one line, saparated in [space]: q w e r t y
This is the original tree
```

```
      y
     w
    t
   r
  q
 e
```

```
This is the tree after reverse
```

```
      e
     q
    r
   t
  w
 y
```

```
//异常处理和健壮性
```

```
//非法数据数量 - 程序结束
```

```
Enter the number of data: qq
Enter the data in one line(only char will be accepted), saparated in [space]:
This is the original tree
```

```
This is the tree after reverse
```

```
//非法数据 - 非法数据 (digit) 被忽略
```

```
Enter the number of data: 6
Enter the data in one line(only char will be accepted), saparated in [space]:
q w e r t 1
invaild input data form
This is the original tree
```

```
      w
     t
    r
   q
  e
```

```
This is the tree after reverse
```

```
      e
     q
    r
   t
  w
```

```
////非法数据 - 非法数据 (string) 只取首字符
```

```
Enter the number of data: 6
Enter the data in one line(only char will be accepted), saparated in [space]:
q w e r t wuuveuuuuew
This is the original tree
```

```
      w
```

```
      w
     / \
    /   \
   /     \
  /       \
 q         t
 /
e

This is the tree after reverse
e
q
  /
  r
   \
   t
    /
   w
  / \
 w   
```

## 作者信息

---

席睿

软工3班

16340247