树形结构Part II

同学们要自己动手，锻炼动手能力！

随机生成二叉树（福利代码）

开课吧

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

typedef struct node {
    char data;
    struct node *lchild, *rchild;
} node;

void preorder(node *p) {
    printf("%c", p->data);
    if (p->lchild != NULL) preorder(p->lchild);
    if (p->rchild != NULL) preorder(p->rchild);
}

void inorder(node *p) {
    if (p->lchild != NULL) inorder(p->lchild);
    printf("%c", p->data);
    if (p->rchild != NULL) inorder(p->rchild);
}

void postorder(node *p) {
    if (p->lchild != NULL) postorder(p->lchild);
    if (p->rchild != NULL) postorder(p->rchild);
    printf("%c", p->data);
```

```
}

void level(node *p) {
    node *que[35];
    int front = 0, rear = 1;
    que[0] = p;
    while (front != rear) {
        node *temp = que[front];
        front++;
        printf("%c", temp->data);
        if (temp->lchild != NULL) que[rear++] = temp->lchild;
        if (temp->rchild != NULL) que[rear++] = temp->rchild;
    }
    printf("\n");
}

node *insert_node(node *p, char c) {
    if (p == NULL) {
        node *t = (node *)malloc(sizeof(node));
        t->data = c;
        t->lchild = t->rchild = NULL;
        return t;
    }
    int ind = rand() % 2;
    if (ind == 0) {
        p->lchild = insert_node(p->lchild, c);
    } else {
        p->rchild = insert_node(p->rchild, c);
    }
    return p;
}

int main() {
```

```
    srand(time(0));
    int n, mark[30] = {0};
    scanf("%d", &n);
    node *root = NULL;
    for (int i = 0; i < n; i++) {
        int x;
        do {
            x = rand() % 26;
        } while (mark[x] == 1);
        mark[x] = 1;
        root = insert_node(root, 'A' + x);
    }
    preorder(root);
    printf("\n");
    inorder(root);
    printf("\n");
    postorder(root);
    printf("\n");
    level(root);
    return 0;
}
```