HW6 林宸昊 PB20000034

6.4

- 若设层数为i，则每层结点数为k ^ (i - 1);
- 父结点编号为(p + k - 2) / k(不考虑不存在情况);
- 第一个儿子编号为p * k, 则第i个为p * k + (i - k + 1)
- (p - 1) % k != 0, 此时兄弟编号为p + 1.

6.5

- 最大深度为n, 最小深度为完全k叉树时的深度，为[log_k(n)] + 1.

6.35

- 首先，由于使用完全二叉树得顺序存储结构并且只考察下标，可以认为考察得就是完全二叉树得对应节点；
- 其次，每往下一层，原二进制序列左移一位，即为乘2，若新增一位为0，代表是左节点，对应为2i；若新增一位为1，代表是右节点，对应为2i+1，正好与二叉树顺序结构的编号方式相对应；
- 因此，二进制序列所代表整数即为下标。若储存在数组中，以0号单元代表头结点。

```
int b_to_d(SqBiTree T, int i){
    if(T[i]) return i;
    else return -1;
}
```

6.45

- 首先编写删除某结点及其所有子结点的算法

```
BiTree FreeBitree(BiTree t){
    if(t){
        FreeBitree(t->lchild);
        FreeBitree(t->rchild);
        free(t);
        t = NULL;
    }
    return t;
}
```

- 再编写按元素删除算法

```
BiTree Del_x(BiTree t, char x){
    if(t){
```

```
        if(t->data == x){
            t = FreeBitree(t);
        }
        else {
            t->lchild = Del_x(t->lchild, x);
            t->rchild = Del_x(t->rchild, x);
        }
    }
    return t;
}
```