

算法 C（按位置插入平衡树形）

//给定表示成一棵平衡二叉树的线性表，变元 k 和指向一个新结点的指针 q，本算法把新结点 q

//插入到树形中（在中根次序下的）第 k 个结点之前（作为新的第 k 个结点）

```
template <class T>
void AVLTree<T>::InsertByPos(int k,AVLNode<T>*q)
{
    AVLNode<T>* t=head;
    AVLNode<T>* s=head->rlink;
    AVLNode<T>* p=head->rlink;
    AVLNode<T>* r;
    int u,m;
    u=k; m=k;
    while(true){
        if(m<=p->RANK) {
            p->RANK++;
            r=p->llink;
            if(r==NULL) {p->llink=q; break;}
            else{
                if(r->balance!=0) {
                    t=p; s=r; u=m;
                }
                p=r;
            }
        }
        else{
            m=p->RANK; r=p->rlink;
            if(r==NULL) {p->rlink=q; break;}
            else{
                if(r->balance!=0) {
                    t=p; s=r; u=m;
                }
                p=r;
            }
        }
    }
    //插入
    q->RANK=1; q->llink=NULL; q->rlink=NULL; q->balance=0;
    //调整平衡系数
    m=u;
    if(m<s->RANK) {r=s->llink; p=r; a=-1;}
    else {r=s->rlink; p=r; a=1; m=s->RANK;}
    while(p!=q)
    {
        if(m<p->RANK) { p->balance=-1; p=p->llink;}
        if(m>p->RANK) { p->balance=1; m=p->RANK; p=p->rlink;}
    }
    if(s->balance==0) {s->balance=a;head->llink=head->llink+1; return;}
    if(s->balance==a) {s->balance=0; return;}
    if(s->balance==a){
        if(r->balance==a){
            p=r;
            *link(a,s)=*link(-a,r);
            *link(-a,r)=s;
            s->balance=0; r->balance=0;
            //更新字段 RANK
            if(a==1) r->RANK=r->RANK+s->RANK;
            if(a==-1) s->RANK=s->RANK-r->RANK;
        }
        else if(r->balance==a){
            p=*link(-a,r);
            *link(a,r)=*link(a,p);
            *link(a,p)=r;
            *link(a,s)=*link(-a,p);
            *link(-a,p)=s;
            if(p->balance==a) {s->balance=-a; r->balance=0;}
            if(p->balance==0) {s->balance=-0; r->balance=0;}
        }
    }
}
```

```
        if(p->balance==a) {s->balance=0;  r->balance=a;}
        p->balance=0;
        if(a==1) {r->RANK=r->RANK-p->RANK; p->RANK=p->RANK-s->RANK;}
        if(a==1){p->RANK=p->RANK+r->RANK; s->RANK=s->RANK-p->RANK;}
    }
    if(s==t->rlink) t->rlink=p;
    else t->llink=p;
}
}
```