简介

zkw线段树是一种改良的非递归式的线段树，发明人张昆玮。

zkw线段树有如下优点：（1）代码简短；（2）非递归；（3）常数小。

缺点：不是很灵活。

操作

建树

自底向上建树。输入了叶子节点的信息任何再一路向上传递信息到根节点。将线段树开成满二叉的形式，那么第一个叶子节点的位置刚好是2^k

```
#define pushup(x) (tree[x]=tree[lson(x)]+tree[rson(x)])

inline void Build() {
    for(M=1;M<N;M<<=1);
    for(int i=M+1;i<=M+N;++i) scanf("%d", &tree[i]);
    for(int i=M-1;i;--i) pushup(i);
}
```

单点修改

```
void modify(int x,int k){
    for(x+=N;x;x>>=1) tree[x]+=k;
}
```

区间查询

```
int query(int l,int r) {
    int ans = 0;
    // l=l+M-1->将查询区间改为L-1;r=r+M+1->将查询区间改为R+1
    // l^r^1 -> 相当于判断l与r是否是兄弟节点
    for(l=l+M-1,r=r+M+1;l^r^1;l>>=1,r>>=1) {
        if(~l&1) // l % 2 == 0 即l是l/2的左儿子
            ans += tree[l^1];
        if(r&1) // r % 2 == 1 即r是r/2的右儿子
            ans += tree[r^1];
    }
    return ans;
}
```

区间修改

标记永久化。

```
void update(int s,int t,int k){
    int lnum=0,rnum=0,nnum=1;
    //lnum:s一路走来已经包含了多少个数
    //rnum:t一路走来已经包含了多少个数
    //nnum:本层每个节点包含几个数
    for(s=N+s-1,t=N+t+1;s^t^1;s>>=1,t>>=1,nnum<<=1){
        tree[s]+=k*lnum;
        tree[t]+=k*rnum;
        if(~s&1){
            add[s^1]+=k;
            tree[s^1]+=k*nnum;
            lnum+=nnum;
        }
        if(t&1){
            add[t^1]+=k;
            tree[t^1]+=k*nnum;
            rnum+=nnum;
        }
    }
    for(;s;s>>=1,t>>=1){
        tree[s]+=k*lnum;
        tree[t]+=k*rnum;
    }
}

int query(int s,int t){ //区间查询也要做出修改
    int lnum=0,rnum=0,nnum=1,ans=0;
    for(s=n+s-1,t=n+t+1;s^t^1;s>>=1,t>>=1,nnum<<=1){
        //根据标记更新
        if(add[s]) ans+=add[s]*lnum;
        if(add[t]) ans+=add[t]*rnum;
        //常规求和
        if(~s&1){
            ans+=tree[s^1];
            lnum+=nnum;
        }
        if(t&1){
            ans+=tree[t^1];
```

```
            rnum+=nnum;
        }
    }
    for(;s;s>>=1,t>>=1){
        ans+=add[s]*lnum;
        ans+=add[t]*rnum;
    }
    return ans;
}
```

其他

zkw线段树还有一个进化体叫cf_zkw线段树，只需要开两倍的空间哦。

zkw：树状数组究竟是什么？就是省掉一半空间后的线段树加上中序遍历

参考资料

https://www.cnblogs.com/Judge/p/9514862.html

https://zhuanlan.zhihu.com/p/29876526

https://baijiahao.baidu.com/s?id=1611019207439457255&wfr=spider&for=pc

https://blog.csdn.net/unicornt_/article/details/52078337

https://blog.csdn.net/keshuqi/article/details/52205884