数据结构

北京大学 洪华敦

线段树基本区间操作

- 1. 单点修改，区间询问：对于每个线段树上的区间维护一个区间的信息之和
- 例子：单点修改，询问区间和，区间最大值
- 2. 区间修改，单点询问：每次修改时，对区间打上一个修改标记，在询问和修改时，一旦访问到一个区间，就将他的标记下传
- 例子：区间加，求单点的值

线段树与标记下传

- 区间修改，区间询问
- 也是通过打标记的方法来进行维护

```
void modify(int me,int l,int r,int x,int y,tag v){
    if(l!=r)down(me);
    if(x<=l&&r<=y){
        giveTag(me,v);
        return;
    }
    int mid=(l+r)>>1;
    if(x<=mid)modify(me<<1,l,mid,x,y,v);
    if(y>mid)modify(me<<1|1,mid+1,r,x,y,v);
    res[me]=mergeData(res[me<<1],res[me<<1|1]);
}
data query(int me,int l,int r,int x,int y){
    if(l!=r)down(me);
    if(x<=l&&r<=y)return res[me];
    data ans=data();
    int mid=(l+r)>>1;
    if(x<=mid)ans=mergeData(ans,res[me<<1]);
    if(y>mid)ans=mergeData(ans,res[me<<1|1]);
}
```

线段树与标记下传

```
struct tag;
struct data;
data res[N<<2];tag lazy[N<<2];
void giveTag(int me,tag v){
    lazy[me]=mergeTag(lazy[me],v);
    res[me]=apply(res[me],v);
}
void down(int me){
    giveTag(me<<1,lazy[me]);
    giveTag(me<<1|1,lazy[me]);
    lazy[me]=tag();
}
```

-

线段树与标记下传

- 可以发现，区间修改区间询问的线段树重点需要实现的是：

  - 标记之间的合并
  - 区间信息之间的合并
  - 区间信息与标记的合并

- 只要能快速地进行以上三个操作，理论上标记和区间信息可以是任何东西

线段树基础题1

- 有一个序列 a[1…n]，要求支持区间加，区间乘，询问区间和

- n,Q<=10^5

线段树基础题2

- 给定 a[1…n]，每次询问给定 K,X，求最大的 i，使得 i<=K 且 a[i]>=X

- n,Q<=10^5

线段树例题1

- 给定序列 a[1…n]，Q 次询问，要求支持：
- 区间乘 -1
- 求区间最大子段和
- n,Q<=10^5

线段树例题2

- 给定序列 a[1…n]，支持单点修改，每次求区间单调栈大小
- n,Q<=100000

线段树例题3

- 有一个序列 a[1…n]，每个 a[i] 是 (c,x,y)，表示颜色和坐标
- 现在支持单点修改以及区间询问：颜色不同的曼哈顿距离最大的一对点的距离
- N,Q<=100000

线段树例题4

- 给定 a[1…n]，要求支持单点修改，以及区间询问 a[L…R] 不能组成的最小的数
- n,Q<=200000

线段树例题5

- 给定 a[1…n] 要求支持：
- 1. 区间加
- 2. 区间变成 max(a[i]+v,0)，v可以是正数也可以是负数
- 3. 求单点当前值
- 4. 求单点历史最大值
- 5. 区间覆盖
- n,Q<=100000

线段树例题6

- 给定序列 a[1…n]，要求支持区间和以及让 a[L…R] 对 x 取 min
- n,Q<=100000

线段树例题7

- 给定序列 a[1…n] 以及k,d，求一个最长的区间，使得最多加入 k 个数后sort后是一个公差为 d 的等差数列
- n<=200000

线段树例题8

- 给定序列 a[1…n]，Q 次询问 a[l…r] 中相差最小的两个数的差
- n,Q<=200000

线段树例题9

- 给定 n，定义 work(x,y) 等于在 [1,n] 的线段树上对 [x,y] 进行区间询问后访问到的点的个数
- 给定 L,R，求 work(i , j) 的和，其中 L<=i<=j<=R
- n,Q<=100000

线段树例题10

- 有 Q 次操作，每次给定 L,R,K,B：对于 L<=i<=R 令 a[i]=max(a[i],K*i+B)
- n,Q<=100000

线段树例题11

- 定义 Min-Max 树是一棵二叉树，每个叶子节点有一个权值
- 现在定义每个非叶子的权值为：有 p 的概率是两个儿子的权值的 max，有 1-p 的概率是两个儿子的权值的 min
- 对于所有可能的 i，输出根节点权值为 i 的概率
- n<=500000

线段树例题12

- 给一棵 n 个点的树，点有颜色，每次询问点 x 子树内距离 x 不超过 d 的点有多少种不同的颜色
- n<=100000

线段树例题13

- 给定数组 a[1…n]，一开始都为1，要求支持以下3种操作：
- 1. 给定 L,R，对于 L<=i<=R，令 a[i]=phi(a[i])
- 2. 给定 L,R,x，对于 L<=i<=R，令 a[i]=x
- 3. 求区间和

线段树例题14

- 给定 n 个区间，以及每个点的价值 val[1…M]，对于每个区间可以选择里面的一个点 i，获得价值 val[i]，每个点最多只能被选一次，求最大价值

- n,M<=5000

线段树例题15

- 有 n 个点，需要支持两种操作：
- 1. 添加一条边，保证加完后还是森林
- 2. 给定 (u,v)，保证 (u,v) 是一条存在的边，求有多少点对 (x,y) 经过了边 (u,v)

线段树例题16

- 给定一个数组 a[1…n]，首先有 Q 次操作，每次会将一个区间升序排序或者降序排序
- 现在你需要求操作后 a[K] 是啥

线段树例题17

火山哥在上小学数学时学习到了分式的求值方法，然后他注意到了一个很有意思的现象：$\frac{a}{\frac{b}{c}}$(即 $a/(b/c)$) 和 $\frac{\frac{a}{b}}{c}$(即$(a/b)/c$) 的值可能是不一样的！于是他开始思考，一个有 n 条横线的分式有什么有趣的性质。

我们可以用一个序列 $a[0...n]$ 和一个 $1...n$ 的排列 $p[1...n]$ 来定义一个 n 重分式，其中 $p[i]$ 表示第 i 条除法横线的长度，$a[i]$ 表示在第 $i+1$ 条横线上面的数的值，其中 $a[n]$ 表示最下面的数字的值。

例如 2重分式 $a/(b/c)$ 用上面方法来定义的话就是 $[a,b,c]$ 和 $[2,1]$

我们定义 $f(a,p)$ 表示该多重分式的值，例如 $f([1,2,3],[2,1])=3/2$，$f([1,2,3],[1,2])=1/6$，$f([2,3,4,5],[2,3,1])=5/6$

现在火山哥想知道的是，在给定 $a[0...n]$ 和 $p[1...n]$ 的情况下，如果每次给定 $1\le l\le r\le n$，那么是否能快速算出 $f(a[l-1...r],p[l...r])$

由于分数有精度问题，你只需要输出答案对 998244353 取模后的值。

Treap介绍

- treap=tree+heap，利用 heap 的性质来维护树高的性质
- 当合并两棵 treap S,T 时，可以根据 size[S] 和 size[T] 来随机决定谁做父亲

Treap合并

```
Treap *Merge(Treap *A,Treap *B){//合并操作
    if(!A)return B;
    if(!B)return A;
    if(A->fix<B->fix){
        A->r=Merge(A->r,B);
        A->updata();
        return A;
    }else{
        B->l=Merge(A,B->l);
        B->updata();
        return B;
    }
}
```

Treap分裂

```
Droot Split(Treap *x,int k){//拆分操作
    if(!x)return Droot(NULL,NULL);
    Droot y;
    if(Size(x->l)>=k){
        y=Split(x->l,k);
        x->l=y.second;
        x->updata();
        y.second=x;
    }else{
        y=Split(x->r,k-Size(x->l)-1);
        x->r=y.first;
        x->updata();
        y.first=x;
    }
    return y;
}
```

Treap其他操作

- 插入：先找到是第几大，然后先分裂再合并，或者直接讨论出一个插入的算法
- 删除：分裂成三部分，然后再合并
- 寻找第 k 大，与 split 类似
- 区间操作：先分裂成三部分把区间提取出来，然后操作
- 可持久化：因为treap没有父亲，所以很简单就能可持久化

平衡树例题1

- 给定一张拓扑图，每条边上都有一个数，且这些数互不相同
- 现在有 Q 次询问，每次给定 x,k，求从 x 出发的字典序第 k 小的路径的长度
- n,Q<=10^5，k<=10^9

平衡树例题2

- 给定一个字符串数组 s[1…n]，要求支持以下操作：
- 1. 给定 L,R,T，对于 L<=i<=R，s[i]=T+s[i]
- 2. 给定 L,R，求lcp(S[L…R]) 的长度
- n,Q<=50000，sum(|T|)<=100000

平衡树例题3

- 有 n 个区间 $[l_i,r_i]$，求一个长为 n 的序列，满足 $l_i<=x_i<=r_i$，且最长上升子序列长度最长

树例题1

- 链加，子树询问，链询问

树例题2

- 给定一棵树，有 Q 次操作：
- 1. 给定 v,d，等概率随机一个点 r，对于所有满足路径 (r,u) 包含 v 的点u，令 a[u]+=d
- 2. 询问 a[u] 的期望值
- n,Q<=200000

树例题3

- 给定一棵带点权的有根树，要求支持单点修改以及求以x为根的子树的最大连通子块和

树例题4

- 给定一棵 n 个点的带点权的树，要求支持以下操作：
- 1. 求一条链上的点权的 gcd
- 2. 链加
- n,Q<=50000

长链剖分

- 每次重儿子选最深的那个
- 应用：O(nlogn+Q) 求出点 x 的第 k 个祖先

动态最小生成树

- 每次加边，要求维护最小生成树
- 对于加的边，强行加进去后会形成一个环，那么我们肯定删掉环上最大的边
- 可以用 LCT 维护

LCT 例题1

- 给定一棵 n 个点的有根树，每个点一开始都有一个互不相同的权值，要求支持以下操作：
- 1. 把 i 到根路径上的所有节点的权值变成一种全新的权值
- 2. 询问 i 子树中所有节点到根路径上不同权值个数的平均值
- n,Q<=200000

LCT与SAM

- 给定一个串 S，每次询问一个区间里有几个不同的子串
- n,Q<=100000

杂题1

- 给定一棵 n 个点的树，每个点有颜色，要求支持修改单点颜色和询问某个颜色的虚树大小
- n,Q<=100000

杂题2

- 给定一个排列 p[1…n]，每次操作可以交换相邻两个数，求最少操作几次可以让这个排列变成山峰型排列
- n<=100000

杂题3

- 给定一堆点，你需要用一些不相交的底边在 x 轴上的矩形覆盖它们
- 一个 w x h 的矩形的代价是 (w+k)*h，求覆盖所有点的最小代价
- n<=10^5