ICPC Template Manual

作者: 贺梦杰

June 13, 2019

Contents

Chapter 1

基础

Chapter 2

搜索

Chapter 3

动态规划

Chapter 4

字符串

4.1 字符串测试

字符串测试内容

Chapter 5

数据结构

5.1 线段树

5.1.1 基础操作

```
const int N = 1e5 + 10;
#define ls(a) (a << 1)
#define rs(a) (a << 1 | 1)
struct node
{
    int val;
    int lazy;
};
node tree[N << 2];
int a[N];
void PushUp(int rt)
{
    tree[rt].val = tree[ls(rt)].val + tree[rs(rt)];
}
void PushDown(int ls, int rs, int rt)
{
    tree[ls(rt)].val += ls * tree[rt].lazy;
    tree[rs(rt)].val += rs * tree[rt].lazy;
    tree[ls(rt)].lazy += tree[rt].lazy;
    tree[rs(rt)].lazy += tree[rt].lazy;
    tree[rt].lazy = 0;
}
void Build(int left, int right, int rt)
{
    if (left == right)
    {
        tree[rt].val = a[left];
        return;
    }
    int mid = (left + right) >> 1;
    Build(left, mid, ls(rt));
    Build(mid + 1, right, rs(rt));
    PushUp(rt);
    //向上更新
}
```

5.1.2 单点更新

```
void Update(int left, int right, int rt, int pos, int val)
{
    if (left == right && left == pos)
    {
        tree[rt].val += val;
        return;
    }
    int mid = (left + right) >> 1;
    if (tree[rt].lazy)
    {
        PushDown(mid - left + 1, right - mid, rt);
    }
    if (mid >= pos)
        Update(left, mid, ls(rt), pos, val);
    else if (pos > mid)
        Update(mid + 1, right, rs(rt), pos, val);
    PushUp(rt);
}
```

例题：*https://www.luogu.org/problemnew/show/P3372*

5.1.3 区间更新

```
void Update(int left, int right, int rt, int s, int t, int val)
{
    if (left >= s && right <= t)
    {
        tree[rt].val += (right - left + 1) * val;
        tree[rt].lazy += val;
        return;
    }
    int mid = (left + right) >> 1;
    if (tree[rt].lazy)
    {
        PushDown(mid - left + 1, right - mid, rt);
    }
    if (mid < s)
        Update(mid + 1, right, rs(rt), s, t, val);
    else if (mid >= t)
        Update(left, mid, ls(rt), s, t, val);
    else
    {
        Update(left, mid, ls(rt), s, t, val);
        Update(mid + 1, right, rs(rt), s, t, val);
    }
    PushUp(rt);
}
```

5.1.4 区间查询

```
void Query(int left, int right, int s, int t, int rt)
{
    if (left >= s && right <= t)
    {
        return tree[rt].val;
    }
    int mid = (left + right) >> 1;
    if (tree[rt].lazy)
        PushDown(mid - left + 1, right - mid, rt);
    long long sum = 0;
    if (mid < s)
        sum += Query(mid + 1, right, rs(rt), s, t, val);
    else if (mid >= t)
        sum += Query(left, mid, ls(rt), s, t, val);
    else
    {
        sum += Query(left, mid, ls(rt), s, t, val);
        sum += Query(mid + 1, right, rs(rt), s, t, val);
    }
    return sum;
}
```

例题：*https://www.luogu.org/problemnew/show/P3373*

5.2 树状数组

推荐阅读：*https://www.cnblogs.com/RabbitHu/p/BIT.html*

5.2.1 单点修改，区间查询

```
#define N 1000100
long long c[N];
int n,q;
int lowbit(int x)
{
    return x&(-x);
}
void change(int x,int v)
{
    while(x<=n)
    {
        c[x]+=v;
        x+=lowbit(x);
    }
}
long long getsum(int x)
{
    long long ans=0;
    while(x>=1)
    {
        ans+=c[x];
        x-=lowbit(x);
    }
    return ans;
}
```

例题：*https://loj.ac/problem/130*

5.2.2 区间修改，单点查询

引入差分数组来解决树状数组的区间更新

```
//初始化
change(i,cur-pre);
//区间修改
change(l,x);
change(r+1,-x);
//单点查询
```

```
getsum(x)
```

例题：*https://loj.ac/problem/131*

5.2.3 区间修改，区间查询

```
//初始化
change(c1,i,cur-pre);
change(c2,i,i*(cur-pre));
//为什么这么写？ 你需要写一下前缀和的表达式
//区间修改
change(c1,l,x);
change(c2,l,l*x);
change(c1,r+1,-x);
change(c2,r+1,-(r+1)*x);
//区间查询
temp1=l*getsum(c1,l-1)-getsum(c2,l-1);
temp2=(r+1)*getsum(c1,r)-getsum(c2,r);
ans=temp2-temp1
```

例题：*https://loj.ac/problem/132*

5.3 二维树状数组

5.3.1 单点修改，区间查询

```
#define N 5050
long long tree[N][N];
long long n,m;
long long lowbit(long long x)
{
    return x&(-x);
}
void change(long long x,long long y,long long val)
{
    long long init_y=y;
    //这里注意n,m的限制
    while(x<=n)
    {
        y=init_y;
        while(y<=m)
        {
            tree[x][y]+=val;
            y+=lowbit(y);
        }
        x+=lowbit(x);
    }
}
long long getsum(long long x,long long y)
{
    long long ans=0;
    long long init_y=y;
    while(x>=1)
    {
        y=init_y;
        while(y>=1)
        {
            ans+=tree[x][y];
            y-=lowbit(y);
        }
        x-=lowbit(x);
    }
    //这里画图理解
    return ans;
}
//初始化
change(x,y,k);
//二维前缀和
ans = getsum(c,d)+getsum(a-1,b-1)-getsum(a-1,d)-getsum(c,b-1);
```

例题: *https://loj.ac/problem/133*

5.3.2 区间修改，区间查询

```
#define N 2050
long long t1[N][N];
long long t2[N][N];
long long t3[N][N];
long long t4[N][N];
long long n,m;
long long lowbit(long long x)
{
    return x&(-x);
}
long long getsum(long long x,long long y)
{
    long long ans=0;
    long long init_y=y;
    long long init_x=x;
    while(x>=1)
    {
        y=init_y;
        while(y>=1)
        {
            ans+=(init_x+1)*(init_y+1)*t1[x][y];
            ans-=(init_y+1)*t2[x][y];
            ans-=(init_x+1)*t3[x][y];
            ans+=t4[x][y];
            y-=lowbit(y);
        }
        x-=lowbit(x);
    }
    return ans;
}
void change(long long x,long long y,long long val)
{
    long long init_x=x;
    long long init_y=y;
    while(x<=n)
    {
        y=init_y;
        while(y<=m)
        {
            t1[x][y]+=val;
            t2[x][y]+=init_x*val;
            t3[x][y]+=init_y*val;
            t4[x][y]+=init_x*init_y*val;
            y+=lowbit(y);
        }
        x+=lowbit(x);
    }
}
//区间修改
change(c+1,d+1,x);
change(a,b,x);
change(a,d+1,-x);
change(c+1,b,-x);
//区间查询
ans=getsum(c,d)+getsum(a-1,b-1)-getsum(c,b-1)-getsum(a-1,d);
```

例题：*https://loj.ac/problem/135*

Chapter 6

图论

Chapter 7

数学

Chapter 8

计算几何

Chapter 9

其他