

前言

用处与一维线段树类似，实现区间的修改、查询工作，知识是在二维的区域中实现的。

两种实现方式

- 普通列表项目每一个节点代表一颗线段树



- 普通列表项目四分法转为一维线段树



复杂度

空间复杂度都是 $O(n^2 \times \log n^2)$ 第一种方法单次操作时间复杂度为 $O(\log n^2)$ 第二种方法最差可以退化到 $O(n)$ 这里介绍第一种方法。思想主要是在二维的区间中将包含目标小区间的区间全部修改，类似于一维情况。

操作

单点修改+矩阵查询

修改

```
void changex(int kx,int l,int r)
{
    changey(kx,1,1,h);
    if(l==r) return;
    int mid=l+r>>1;
    if(x<=mid) changex(kx<<1,l,mid);
    else changex(kx<<1|1,mid+1,r);
}
void changey(int kx,int ky,int l,int r) //方法一
{
    sum[kx][ky]++;
    if(l==r) return;
    int mid=l+r>>1;
    if(y<=mid) changey(kx,ky<<1,l,mid);
    else changey(kx,ky<<1|1,mid+1,r);
}
```

```
void changey(int kx,int ky,int l,int r) //方法二
{
    if(l==r)
    {
        sum[kx][ky]++;
        return;
    }
    int mid=l+r>>1;
    if(y<=mid) changey(kx,ky<<1,l,mid);
    else changey(kx,ky<<1|1,mid+1,r);
    sum[kx][ky]=sum[kx][ky<<1]+sum[kx][ky<<1|1];
}
```

查询

```
void queryy(int kx,int ky,int l,int r)
{
    if(l>=yl && r<=yr)
    {
        cnt+=sum[kx][ky];
        return;
    }
    int mid=l+r>>1;
    if(yl<=mid) queryy(kx,ky<<1,l,mid);
    if(yr>mid) queryy(kx,ky<<1|1,mid+1,r);
}

void queryx(int kx,int l,int r)
{
    if(l>=xl && r<=xr)
    {
        queryy(kx,1,1,h);
        return;
    }
    int mid=l+r>>1;
    if(xl<=mid) queryx(kx<<1,l,mid);
    if(xr>mid) queryx(kx<<1|1,mid+1,r);
}
```

矩阵修改+单点查询

修改

```
void changey(int kx,int ky,int l,int r)
{
    if(y1<=l&&r<=y2)
    {
```

```

        sum[kx][ky]++;
        return;
    }
    int mid=l+r>>1;
    if(y1<=mid) changey(kx,ky<<1,l,mid);
    if(y2>mid) changey(kx,ky<<1|1,mid+1,r);
}
void changex(int kx,int l,int r)
{
    if(x1<=l&&r<=x2)
    {
        changey(kx,1,1,n);
        return;
    }
    int mid=l+r>>1;
    if(x1<=mid) changex(kx<<1,l,mid);
    if(x2>mid) changex(kx<<1|1,mid+1,r);
}

```

查询

```

void queryy(int kx,int ky,int l,int r)
{
    ans+=sum[kx][ky];
    if(l==r) return;
    int mid=ly+ry>>1;
    if(y<=mid) queryy(kx,ky<<1,l,mid);
    else queryy(kx,ky<<1|1,mid+1,r);
}
void queryx(int kx,int l,int r)
{
    queryy(kx,1,1,n);
    if(l==r) return;
    int mid=l+r>>1;
    if(x<=mid) queryx(kx<<1,l,mid);
    else queryx(kx<<1|1,mid+1,r);
}

```

参考

<https://www.cnblogs.com/TheRoadToTheGold/p/8151375.html>

From: <https://wiki.cvbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbacm.com/doku.php?id=2020-2021:teams:no_morning_training:shaco:%E7%9F%A5%E8%AF%86%E7%B2%B9:%E6%95%B0%E6%BD%AE%E7%BB%93%E6%9E%84:%E4%BA%8C%E7%BB%B4%E7%BA%BF%E6%AE%B5%E6%A0%91

Last update: 2020/06/05 15:49

