目录

1 基本操作

1.1 离散化

1.1.1 一维线离散化

```
struct LINE{
    int l,r;
}line[SIZE*16];
int lisan[SIZE*16];
void Discrete(int N){//bl存的是编号为[1,N]的线,在每个线段的尾部插入一个断点
    int lisantot=0;
    for(int i=1;i<=N;i++){
        lisan[lisantot++]=line[i].l;
        lisan[lisantot++]=line[i].r;
        lisan[lisantot++]=line[i].r+1;
    }
    sort(lisan,lisan+lisantot);
    int lisanlen=unique(lisan,lisan+lisantot)-lisan;
    for(int i=1;i<=N;i++){
        line[i].l=lower_bound(lisan,lisan+lisanlen,line[i].l)-lisan+1;
        line[i].r=lower_bound(lisan,lisan+lisanlen,line[i].r)-lisan+1;
    }
}
```

1.1.2 二维点离散化

```
#define ll long long
const int SIZE=1e5;
struct point{
    ll x,y;
}p[SIZE];
bool cmp_x(point a,point b){return a.x < b.x;}
bool cmp_y(point a,point b){return a.y < b.y;}
void Discrete(int n){//n个点,下标[1,n]
    sort(p+1,p+n+1,cmp_x);
    int last=p[1].x,num=1;
    p[1].x = num = 1;
    for(int i=2;i<=n;i++){
        if(p[i].x == last)p[i].x=num;
        else{
            last = p[i].x;
            p[i].x=++num;
        }
```