扫描线

```
#include <bits/stdc++.h>
using namespace std;

typedef long long ll;
const int N = 1e5 + 10;
struct Segment{
    ll x, y1, y2, k;
    bool operator< (const Segment &t)const{
        return x < t.x;
    }
}seg[N * 2];
struct Node{
    int l, r, cnt;
    ll len;
}tr[N * 8];
vector<ll> ys;
int find(ll y){
    return lower_bound(ys.begin(), ys.end(), y) - ys.begin();
}

void build(int u, int l, int r){
    tr[u] = {l, r, 0, 0};
    if(l == r) return;
    int mid = l + r >> 1;
    build(u << 1, l, mid), build(u << 1 | 1, mid + 1, r);
}

void pushup(int u){
    if(tr[u].cnt) tr[u].len = ys[tr[u].r + 1] - ys[tr[u].l];
    else if(tr[u].l != tr[u].r){
        tr[u].len = tr[u << 1].len + tr[u << 1 | 1].len;
    }
    else tr[u].len = 0;
}

void modify(int u, int l, int r, ll k){
    if(tr[u].l >= l && tr[u].r <= r){
        tr[u].cnt += k;
        pushup(u);
    }
    else{
        int mid = tr[u].l + tr[u].r >> 1;
        if(l <= mid) modify(u << 1, l, r, k);
        if(mid < r) modify(u << 1 | 1, l, r, k);
        pushup(u);
    }
}

int main(){
    int n;
```

```
    cin >> n;
    for(int i = 0, j = 0; i < n; i++){
        ll x1, y1, x2, y2;
        scanf("%lld%lld%lld%lld", &x1, &y1, &x2, &y2);
        seg[j++] = {x1, y1, y2, 1};
        seg[j++] = {x2, y1, y2, -1};
        ys.push_back(y1), ys.push_back(y2);
    }

    sort(ys.begin(), ys.end());
    ys.erase(unique(ys.begin(), ys.end()), ys.end());

    build(1, 0, ys.size() - 2);

    sort(seg, seg + 2 * n);

    ll ans = 0;
    for(int i = 0; i < n * 2; i++){
        if(i) ans += tr[1].len * (seg[i].x - seg[i - 1].x);
        modify(1, find(seg[i].y1), find(seg[i].y2) - 1, seg[i].k);
    }

    cout << ans;

    return 0;
}
```