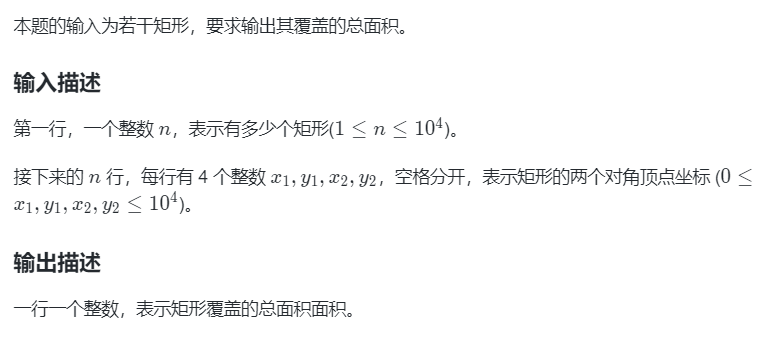
原题：[油漆面积](https://www.lanqiao.cn/problems/105/learning/?page=1&first_category_id=1&second_category_id=3)



扫描线算法模板（无离散化，无线段树）：

#include<bits/stdc++.h>

using namespace std;

int n;

vector<vector<int>> v(10005,vector<int>(4));

struct line{

    int x,y1,y2,f;

    bool operator<(line l)

    {

        return x<l.x;

    }

};

int main()

{

    cin>>n;

    line l[n\*2];

    for(int i=0;i<n;i++)

    cin>>v[i][0]>>v[i][1]>>v[i][2]>>v[i][3];

    for(int i=0;i<n;i++)

    {

        int x1=v[i][0],y1=v[i][1],x2=v[i][2],y2=v[i][3];

        if(x1>x2)

        swap(x1,x2);

        if(y1>y2)

        swap(y1,y2);

        l[i].x=x1,l[i].y1=y1,l[i].y2=y2,l[i].f=1;

        l[i+n].x=x2,l[i+n].y1=y1,l[i+n].y2=y2,l[i+n].f=-1;

    }

    sort(l,l+n\*2);

    vector<int> vv(10005,0);

    int prex=0;

    int ans=0;

    int len=0;

    for(auto ll:l)

    {

        int x=ll.x;

        int y1=ll.y1;

        int y2=ll.y2;

        int f=ll.f;

        int newlen=len;

        if(y1>y2)

        swap(y1,y2);

        for(int y=y1;y<y2;y++)

        {

            vv[y]+=f;

            if(vv[y]==1&&f==1)

            newlen++;

            if(vv[y]==0)

            newlen--;

        }

        ans+=len\*(x-prex);

        prex=x;

        len=newlen;

    }

    cout<<ans;

}

扫描线算法时间复杂度：

在遍历每条扫描线后，我们会更改len的长度，这是一个区间修改的过程，如果使用线段树，时间复杂度会降为

如果y的范围是1e9而x点的数量只有1e5，区间修改的时间和空间复杂度会大额增加，我们需要离散化