Pastebin link: http://paste.ubuntu.com/24705577/

#define SIZE1 100105

#define SIZE2 10000105

#define callLeft s,mid,nd+nd

#define callRight mid+1,e,nd+nd+1

using namespace std;

int n, q;

int fre[SIZE2], segTree[4\*SIZE2], ara[SIZE1];

struct querystruct{

int l, r, x1, x2;

int val[4];

}qara[100105];

struct subquery{

int v, x, qno, type;

void subq(int val, int xx, int qq, int tt){

v=val;

x=xx;

qno=qq;

type=tt;

}

};

subquery sqara[4\*SIZE1];

bool cmp(subquery a, subquery b){

return a.v<b.v;

}

void build(int s, int e, int nd){

if(s==e){

segTree[nd]=fre[s];

return;

}

int mid=(s+e)>>1;

build(callLeft);

build(callRight);

segTree[nd]=segTree[nd+nd]+segTree[nd+nd+1];

}

void update(int s, int e, int nd, int pos){

if(s>pos || e<pos) return;

if(s==e){

segTree[nd]--;

return;

}

int mid=(s+e)>>1;

update(callLeft, pos);

update(callRight, pos);

segTree[nd]=segTree[nd+nd]+segTree[nd+nd+1];

}

int query(int s, int e, int nd, int x){

if(e<x) return 0;

if(s>=x) return segTree[nd];

int mid=(s+e)>>1;

return query(callLeft, x)+query(callRight, x);

}

void query\_input(int i){

scanf("%d %d %d %d", &qara[i].l, &qara[i].r, &qara[i].x1, &qara[i].x2);

sqara[4\*i+0].subq(qara[i].l, qara[i].x1, i, 0);

sqara[4\*i+1].subq(qara[i].l, qara[i].x2+1, i, 1);

sqara[4\*i+2].subq(qara[i].r+1, qara[i].x1, i, 2);

sqara[4\*i+3].subq(qara[i].r+1, qara[i].x2+1, i, 3);

}

void solve(){

sort(sqara, sqara+4\*q, cmp);

memset(fre, 0, sizeof(fre));

for(int i=0; i<n; i++){

fre[ara[i]]++;

}

build(0, 10000001, 1);

int pos=0;

for(int i=0; i<4\*q; i++){

while(pos<sqara[i].v){

update(0, 10000001, 1, ara[pos]);

pos++;

}

if(pos==n) qara[sqara[i].qno].val[sqara[i].type]=0;

else qara[sqara[i].qno].val[sqara[i].type]=query(0, 10000001, 1, sqara[i].x);

}

for(int i=0; i<q; i++){

printf("%d\n", (qara[i].val[0]-qara[i].val[2])-(qara[i].val[1]-qara[i].val[3]));

}

}