**Segment Tree 2D**

**Tested on LightOj 1081 - Square Queries**

int mx[MX\*4][MX\*4], n;

void inserty(int idx, int stx, int edx, int idy, int sty, int edy, int sy, int ey, int val)

{

if(sty==sy && edy==ey)

{

if(stx==edx) mx[idx][idy]=val;

else mx[idx][idy]=max(mx[idx<<1][idy], mx[(idx<<1) | 1][idy]);

return;

}

int mid=(sty+edy)>>1, lft=idy<<1, rgt=lft | 1;

if(ey<=mid) inserty(idx, stx, edx, lft, sty, mid, sy, ey, val);

else if(sy>mid) inserty(idx, stx, edx, rgt, mid+1, edy, sy, ey, val);

else

{

inserty(idx, stx, edx, lft, sty, mid, sy, mid, val);

inserty(idx, stx, edx, rgt, mid+1, edy, mid+1, ey, val);

}

mx[idx][idy]=max(mx[idx][lft], mx[idx][rgt]);

return;

}

void insert(int idx, int stx, int edx, int sx, int ex, int sy, int ey, int val)

{

if(stx!=sx || edx!=ex)

{

int mid=(stx+edx)>>1, lft=idx<<1, rgt=lft | 1;

if(ex<=mid) insert(lft, stx, mid, sx, ex, sy, ey, val);

else if(sx>mid) insert(rgt, mid+1, edx, sx, ex, sy, ey, val);

else

{

insert(lft, stx, mid, sx, mid, sy, ey, val);

insert(rgt, mid+1, edx, mid+1, ex, sy, ey, val);

}

}

inserty(idx, stx, edx, 1, 1, n, sy, ey, val);

return;

}

int queryy(int idx, int idy, int sty , int edy, int sy, int ey)

{

if(sty==sy && edy==ey) return mx[idx][idy];

int mid=(sty+edy)>>1, lft=idy<<1, rgt=lft | 1;

if(mid>=ey) return queryy(idx, lft, sty, mid, sy, ey);

else if(sy>mid) return queryy(idx, rgt, mid+1, edy, sy, ey);

else return max(queryy(idx, lft, sty, mid, sy, mid), queryy(idx, rgt, mid+1, edy, mid+1, ey));

}

int query(int idx, int stx , int edx, int sx, int ex, int sy, int ey)

{

if(stx==sx && edx==ex) return queryy(idx, 1, 1, n, sy, ey);

int mid=(stx+edx)>>1, lft=idx<<1, rgt=lft | 1;

if(mid>=ex) return query(lft, stx, mid, sx, ex, sy, ey);

else if(sx>mid) return query(rgt, mid+1, edx, sx, ex, sy, ey);

else return max(query(lft, stx, mid, sx, mid, sy, ey), query(rgt, mid+1, edx, mid+1, ex, sy, ey));

}