<http://paste.ubuntu.com/24464937/>

#define SIZE 100105

#define INF 10000000000000000

typedef long long ll;

struct point{

int x, y, ind;

}ara[SIZE];

bool cmpx(point a, point b){

return a.x<b.x;

}

bool cmpy(point a, point b){

return a.y<b.y;

}

int flag[SIZE];

inline ll sq(const ll &x){

return x\*x;

}

inline ll sqdist(const point &a, const point &b){

return sq(a.x-b.x)+sq(a.y-b.y);

}

int n; //number of points

point X[SIZE]; //points sorted by increasing value of X co-ordinate

point Y[SIZE]; //points sorted by increasing value of Y co-ordinate

inline ll closestPair(point \*X, point \*Y, int n){

if(n==1) return INF;

if(n==2) return sqdist(X[0], X[1]);

int i, j, k, n1, n2, ns, m=n>>1;

point Xm=X[m-1], \*XL, \*XR, \*YL, \*YR, \*YS;

ll lt, rt, dd, tmp;

XL=new point[m], YL=new point[m];

XR=new point[m+1],YR=new point[m+1];

YS=new point[n];

for(i=0; i<m; i++){

XL[i]=X[i];

flag[X[i].ind]=0;

}

for(; i<n; i++){

XR[i-m]=X[i];

flag[X[i].ind]=1;

}

for(i=n2=n1=0; i<n; i++){

if(!flag[Y[i].ind]) YL[n1++]=Y[i];

else YR[n2++]=Y[i];

}

lt=closestPair(XL, YL, n1);

rt=closestPair(XR, YR, n2);

dd=min(lt, rt);

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

if(sq(Y[i].x-Xm.x)<dd) YS[ns++]=Y[i];

}

for(j=0; j<ns; j++){

for(k=j+1; k<ns && sq(YS[k].y-YS[j].y)<dd; k++){

dd=min(dd, sqdist(YS[j], YS[k]));

}

}

delete[] XL;

delete[] XR;

delete[] YL;

delete[] YR;

delete[] YS;

return dd;

}