# [【 HDU2966 】In case of failure（KD-Tree）](https://www.cnblogs.com/flipped/p/6493340.html)

### **题意**

给平面直角坐标系下的n个点的坐标，求离每个点和它最近点的距离的平方。2≤n≤1052≤n≤105

### **题解**

这题是裸的kdtree。  
kdtree就是k-dimension tree的缩写，是一种分割k维数据空间的数据结构，可用来多维空间数据的范围搜索和最近邻搜索。  
这题只是求2维的最近的点，代码比较短。以下是我对算法的理解：

#### **算法过程：**

##### **建树：**

分割区间：区间[L,R]，m为中点。当前分割方式为d，d=0，则纵向分割，按x坐标排序，否则横向分割，按y坐标排序。将中位数放在m位置，小于中位数的放在m前面（不必有序），大于中位数的放在后面（不必有序），这一步可以调用stl的nth\_element函数。

递归分割子区间：分割左右子树时d为d^1。

##### **查询：**

二叉树查找：用a[m]点到p的距离md来更新一下最小距离nd。p.x[i]为第i维的坐标，如果p.x[d]>a[m].x[d]，则查询右子树，否则左子树，直到叶子节点。

回溯：因为最近点未必在p所在的子空间里，因此判断一下p到a[m]所在的分割线的直线距离是否小于当前最小距离nd，若是，则另一半子空间可能存在点距离p更近，于是进入查找。

其它就是，注意最近点不能是本身，nd初始值为0，nd0则一定要更新，否则若md0则不能拿来更新nd。

### **代码**

#include<cstdio>

#include<algorithm>

#define N 100005

#define ll long long

using namespace std;

struct point{

int x[2];}a[N],b[N];

ll nd;int t,n,now;

bool cmp(point a,point b){return a.x[now] < b.x[now];}

ll sqr(int x){return (ll)x\*x;}

ll dis(point a,point b){return sqr(a.x[0]-b.x[0])+sqr(a.x[1]-b.x[1]);}

void build(int l,int r,int d){

if(l>=r) return;

int m=l+r>>1;

now=d;

nth\_element(a+l,a+m,a+r,cmp);

build(l,m,d^1);

build(m+1,r,d^1);}void query(int l,int r,int d,point p){

if(l>=r)return;

int m=l+r>>1,sl=l,sr=m;

ll md=dis(a[m],p);

if(nd==0||md&&nd>md)nd=md;

if(p.x[d]>a[m].x[d])sl=m+1,sr=r;

query(sl,sr,d^1,p);

if(nd>sqr(a[m].x[d]-p.x[d]))

query(l+m+1-sl,m+r-sr,d^1,p);}int main(){

scanf("%d",&t);

while(t--){

scanf("%d",&n);

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

scanf("%d%d",a[i].x,a[i].x+1),b[i]=a[i];

build(0,n,0);

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

nd=0,query(0,n,0,b[i]),printf("%lld\n",nd);

}

return 0;

}