题目大意：给出n条线段两个端点的坐标，问所有线段投影到一条直线上，如果这些所有投影至少相交于一点就输出Yes！，否则输出No！。

解题思路：如果有存在这样的直线，过投影相交区域作直线的垂线，该垂线必定与每条线段相交，问题转化为问是否存在一条线和所有线段相交

直线肯定经过两个端点。

枚举端点，判断直线和线段是否相交。

细节要注意，判断重合点。

还有就是加入只有一条线段的话，刚好直线是过同一条直线的。

所以保险的做法是枚举所有的两个端点，包括同一条直线的。

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <queue>

#include <map>

#include <vector>

#include <set>

#include <string>

#include <math.h>

#include<iomanip>

using namespace std;

typedef long long ll;

const int INF=1000000007;

const double eps = 1e-8;

const double PI = acos(-1.0);

int n ;

int sgn(double x)//判断符号

{

if(fabs(x) < eps)return 0;

if(x < 0)return -1;

else return 1;

}

struct Point

{

double x,y;

Point() {}

Point(double \_x,double \_y)

{

x = \_x;

y = \_y;

}

Point operator -(const Point &b)const

{

return Point(x - b.x,y - b.y);

}

//叉积

double operator ^(const Point &b)const

{

return x\*b.y - y\*b.x;

}

//点积

double operator \*(const Point &b)const

{

return x\*b.x + y\*b.y;

}

//绕原点旋转角度B（弧度值），后x,y的变化

void transXY(double B)

{

double tx = x,ty = y;

x = tx\*cos(B) - ty\*sin(B);

y = tx\*sin(B) + ty\*cos(B);

}

};

struct Line

{

Point s,e;

Line() {}

Line(Point \_s,Point \_e)

{

s = \_s;

e = \_e;

}

//两直线相交求交点

//第一个值为0表示直线重合，为1表示平行，为2是相交

//只有第一个值为2时，交点才有意义

pair<Point,int> operator &(const Line &b)const

{

Point res = s;

if(sgn((s-e)^(b.s-b.e)) == 0)

{

if(sgn((b.e-s)^(b.e-b.s)) == 0)

return make\_pair(res,0);//重合

else return make\_pair(res,1);//平行

}

//相交

double t = ((s-b.s)^(b.s-b.e))/((s-e)^(b.s-b.e));

res.x += (e.x-s.x)\*t;

res.y += (e.y-s.y)\*t;

return make\_pair(res,2);

}

};

Line line[100010];

//\*判断线段相交

bool inter(Line l1,Line l2)//true就是相交

{

return

max(l1.s.x,l1.e.x) >= min(l2.s.x,l2.e.x) &&

max(l2.s.x,l2.e.x) >= min(l1.s.x,l1.e.x) &&

max(l1.s.y,l1.e.y) >= min(l2.s.y,l2.e.y) &&

max(l2.s.y,l2.e.y) >= min(l1.s.y,l1.e.y) &&

sgn((l2.s-l1.e)^(l1.s-l1.e))\*sgn((l2.e-l1.e)^(l1.s-l1.e)) <= 0 &&

sgn((l1.s-l2.e)^(l2.s-l2.e))\*sgn((l1.e-l2.e)^(l2.s-l2.e)) <= 0;

}

double dist(Point a,Point b)

{

return sqrt((a-b)\*(a-b));

}

bool Seg\_inter\_line(Line l1,Line l2) //判断直线l1和线段l2是否相交

{

return sgn((l2.s-l1.e)^(l1.s-l1.e))\*sgn((l2.e-l1.e)^(l1.s-l1.e)) <= 0;

}

bool check(Line l1)

{

if(sgn(dist(l1.s,l1.e))==0)

return 0;

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

if(Seg\_inter\_line(l1,line[i])==0)

return 0;

return 1;

}

int main()

{

//freopen("input.txt","r",stdin);

int T;

cin>>T;

double x1,y1,x2,y2;

while(T--)

{

cin>>n;

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

{

cin>>x1>>y1>>x2>>y2;

line[i]=Line(Point(x1,y1),Point(x2,y2));

}

bool flag=0;

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

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

if(check(Line(line[i].s,line[j].s))||check(Line(line[i].s,line[j].e))||check(Line(line[i].e,line[j].s))||check(Line(line[i].e,line[j].e)))

{

flag=1;

break;

}

if(flag)

puts("Yes!");

else puts("No!");

}

return 0;

}