# Question6,7

On a besoin d'utiliser les <Diviser pour Régner> pour les dernières deux questions. Nous avons fini ce tp par java et python, le python fonctionne bien, java ne passe pas les tests avec de grandes quantités de données. Dans ce tp, on a besoin de comparer la relation entre deux ligne, comparé avec le question 4, on a besoin d'utiliser les ligne comme les objets. Dans la programmation de java, Il y a beaucoup types de relations entre différents segments.

if(y\_prem>y\_deux)

{ if(x2\_prem>x1\_deux){

resultat.add(new imm\_point(x1\_prem, y\_prem, x2\_prem));

if(x2\_prem<x2\_deux){

resultat.add(new imm\_point(x2\_prem, y\_deux, x2\_deux));}}

else{

resultat.add(new imm\_point(x1\_prem, y\_prem, x2\_prem));

resultat.add(new imm\_point(x1\_deux, y\_deux, x2\_deux)); }}

if(y\_prem==y\_deux){

if(x2\_prem>x1\_deux){

if(x2\_prem<=x2\_deux){

{ resultat.add(new imm\_point(x1\_prem,y\_prem,x2\_deux));

}}

else{resultat.add(new imm\_point(x1\_prem,y\_prem,x2\_prem));

}

}

else{

resultat.add(new imm\_point(x1\_prem,y\_prem,x2\_prem));

resultat.add(new imm\_point(x1\_deux,y\_prem,x2\_deux));}

}

if(y\_prem<y\_deux)

{

if(x2\_prem>x1\_prem){

resultat.add(new imm\_point(x1\_prem, y\_prem, x1\_deux));

if(x2\_deux<=x2\_prem){

resultat.add(new imm\_point(x1\_deux, y\_deux, x2\_deux));

resultat.add(new imm\_point(x2\_prem, y\_prem, x2\_deux)); }

else{

resultat.add(new imm\_point(x1\_deux, y\_deux, x2\_deux));}

}

else{

resultat.add(new imm\_point(x1\_prem, y\_prem, x2\_prem));

resultat.add(new imm\_point(x1\_deux, y\_deux, x2\_deux)); }

}

return resultat;

Maintenant on a su quand les deux ligne touches , on va laisser les quels points. On veux utiliser le Algortheme dichotomie, utilise les resultats des une moité de ligne pour comparer avec l'autre partie. Apres bcp de fois de compararion, finalement nous pouvons comparer les deux lignes. Mais parce que la probleme de vide pointer de JAVA, je ne peux pas verifier mon idea.

divide(liste\_imms g, d)

**si** g==d

**return** liste\_imms get(g)

milieu = (g+d)/2

l1 = *divide*(res,g,milieu)

l2 = *divide*(res,milieu+1,d)

**return** *f*(l1,l2)

/\*f est une fonctionne de Question 6 pour comparer les deux ligne et trouver les points utilisant .\*/

Finalement j'utilise une autre fonctionne. J'utilise la nouvelle ligne de comparer avec les anciens résultats.

var a compare b

bool=0

for i=2 to l.length-1

{ res =var;

bool=0

for j=0 to res.length-1

{

if var[j].x2>l.[i].x1{

interm=var[j];

var.remove(var[j]);

var.concat(interm compare l[i])

bool=1}

if (bool=0)

{var.add l[i]}

} } }