# Question4:

Dans cette question, on a besoin de mélanger les points de deux lignes des toits, et juste prendre les points qui sont les plus hauts, c'est-à-dire que on a besoin des fonctions pour comparer les relations entre deux ligne (a side connects with the ground(x==0)). Les codes complètes sont dans notre programmation <question4.java>

on ajoute les points en lignes 1 et lignes 2. Ajoute deux points (-100,-100) pour

assurer l'interator n'a pas erreurs (pointer vide). Et puis on peut faire deux iterators.Et puis on cree deux variables h1,h2 pour conserver le hauter de points

On conserve le premier point de chaque ligne dans chaque iterator, et puis on peut avancer les points par iterator.

while(l1\_next existe, l2\_next existe)

  {

if(pt1.x < pt2.x || (pt1.x == pt2.x && pt1.y > pt2.y)) {

/\*si le x de premier point est plus petits que le x de deuxieme point ou ils sont equals mais ont la meme hauteur \*/

h1 = pt1.y;

//faire le hauteur de premier point equals le hauteur de premier points

    mergeligne.add(new Point(pt1.x, max(h1, h2)));

/\*ajoute le point dans notre final liste de points, x equals le x de premier points, y equals le plus grand hauteur entre h1 et h2 \*/

    pt1 = it1.next();

//advancer le premier point.

    }

else

avancer le point\_suivant

else {

    h2 = pt2.y;

//faire le hauteur de premier point equals le hauteur de deuxieme points

mergeligne.add(new Point(pt2.x, Math.max(h1, h2)));

/\*ajoute le point dans notre final liste de points, x equals le x de premier points, y equals le plus grand hauteur entre h1 et h2 \*/

    pt2 = it2.next(); ///advancer le deuxieme point.

        }

La complexité est en O(n)