```
class Plane:
    def __init__(self, lstx, lsty):
        self.x_min = lstx[0]
        self.x_max = lstx[1]
        self.y_min = lsty[0]
        self.y_max = lsty[1]
        self.x_mid = (self.x_min+self.x_max)/2
        self.y_mid = (self.y_min+self.y_max)/2
        self.q1 = [[self.x_min,self.x_mid],[self.y_min,self.y_mid]]
        self.q2 = [[self.x_mid,self.x_max],[self.y_min,self.y_mid]]
        self.q3 = [[self.x_mid,self.x_max],[self.y_mid,self.y_max]]
        self.q4 = [[self.x_min,self.x_mid],[self.y_mid,self.y_max]]
    def whichQuadrant(self, point):
        x = point[0]
        y = point[1]
        if x>=self.x_min and x<self.x_mid:
            if y>=self.y_min and y<self.y_mid:
                return Plane(self.q1[0],self.q1[1])
            elif y>=self.y_mid and y<self.y_max:
                return Plane(self.q4[0],self.q4[1])
        elif x>=self.x_mid and x<self.x_max:
            if y>=self.y_min and y<self.y_mid:
                return Plane(self.q2[0],self.q2[1])
            elif y>=self.y_mid and y<self.y_max:
                return Plane(self.q3[0],self.q3[1])
        else:
            return None
    def __str__(self):
        return "x_min = "+str(self.x_min)+", x_max = "+str(self.x_max)+"\ny_min = "+
str(self.y_min)+", y_max = "+str(self.y_max)
    def boundingBox(self):
        return [[self.x_min,self.y_min],[self.x_min,self.y_max],[self.x_max,self.y_m
ax],[self.x_max,self.y_min]]
```