Algorithm 1:

In order to know the robot location we used the following properties within the class point message robot and air.

Air will receive Robot and message in order to help set the location of the robot we will make one method that will help the robot know his neighbours, very simply first we will use the message class to send a message then using one of the prior methods we will insert the message location as in algorithm 0, after so we will make a new function in the air class that will set the robot message location as the robot actual location figuring who is neighbours are by checking which robots are the closest using pitagoras sentence of distance between to points while using points x y coordinates in each of them, after figuring whos closest we established neighbours, we can also use the static robots as a towers of verification regarding position for example what we do is when a robot is static and he gets our message.

Now for the actual algorithm ,

We got 2 options to check that first, we make a function that send a message to all robots using the class message, making an array message to all robot IDs except the one in use then, the location of the message received will be saved as a point, we will add in air an simple equation saying the location the message was received is the location of the IDd robot the we will use a simple mathematical equation testing each of the points in a loop from I to number of robots that received the message, we will check the distance using the coordinates of points x,y given to us by method points , simply after we got the point in message location we can reach to the class point to gather the X and Y coordinates in the Ring, We will use the mathematic equation ((x2-x1)^2 + (y2-y2)^2)^0.5 for the distance measurement, we will insert in an array the in a loop one by one the distances and the ids of each robot distance, then we will go use min and max in order to organize the distances from shortest to longest, then we know which ID robots are the closest to us. (our neighbours).

Its important to add the deviation of + - 20% so we take it under account when checking distance therefor if we in need to get a very precise answer we will make a loop that will repeat the process of the messaging and distance a few times in a second or 2 to lower the deviation.

2nd option, we will make a variable that will messure the strength of the signal and use the signal measurement to figure out the point, we decided not to use it due to it being complex, for using messages will give us the location of neighbours when there is even 1 line of signal. While measuring with signal can give us even bigger deviations due to obsticales and mechanic issues.