



```

// Assume that sensor node  $S$  wants to calculate its location
// Input:
//    $S_n$  denotes the set of neighbouring anchors of sensor  $S$ .
//    $S_A$  denotes the set of neighbouring anchors of anchor  $A$ ,
//   where  $A$  is one of neighbouring anchors of sensor  $S$ .
//    $RRS_{AB}$  denotes the signal strength recorded from node  $A$  to node  $B$ .
Process LocationEstimation()
{
    RingSet  $R = \{\}$ ;
    While( $S_n$  has more elements){
        step 1: Anchor  $A = S_n.nextElements()$ ;
        step 2: Split  $S_A$  into two parts:  $S_{A1}$  and  $S_{A2}$ , such that
            each element  $I$  in  $S_{A1}$  has larger  $RRS_{AI}$  than  $RRS_{AS}$  and
            each element  $J$  in  $S_{A2}$  has smaller  $RRS_{AJ}$  than  $RRS_{AS}$ .
        step 3: if ( $(S_A == null)$  or  $(S_{A2} == null)$ ) goto step 1.
             $J$  = the element with the largest value in  $S_{A2}$ ;
             $d_2$  = distance between  $J$  and  $A$ ;
            if ( $S_{A1} == null$ ) {  $d_1 = d_2$ ; }
            else {
                 $I$  = the element with the smallest value in  $S_{A1}$ ;
                 $d_1$  = distance between  $I$  and  $A$ ;
            }
            Generate a ring  $r$  centered at  $A$  with inner radius  $d_1$  and outer radius  $d_2$ ;
             $R = R + \{r\}$ ; // insert  $r$  into  $R$ .
        }
        step 4: Calculate the intersection area of all rings in  $R$ ;
        step 5: return {the gravity of this intersection area};
    }
}

```

