Design Challenge 3: Guide Me Home

1 The Challenge
The robots are a long way from home, with many twists and turns along the way. Thankfully, there are many friendly robots on the path who are happy to lead your robot to safety. Each robot can calculate its position relative to nearby robots, and navigate towards them. The trick is to figure out which robot to follow. To do this, we need hops.

Each guide robot counts its number of hops from the goal robot. By always navigating to a robot with lower hops than itself, your robot will eventually make its way safely home.

2 Follow Guide Behavior
These are the functions needed to complete this design challenge:

1. behControl.guideMsgGetParent(): Chooses the neighbor of with lowest hops to be parent. Input: None Output: parent - Chosen parent robot of this robot.

2. behControl.nbrGetNbrBearing(nbr): Gets the bearing of the robot to the input neighboring robot. Input: nbr - Chosen neighboring robot Output: bearing - the displacement (in radians) from the front of the robot to the neighboring robot.

You will write the function navParent(). We do all the behind the scenes calculations to determine hops and the parent robot, but you will need to use behControl.guideMsgGetParent() to chose a guide nbr to move towards. navParent() will also use nbrGetNbrBearing(nbr). You will need to implement if, elif, and else statements to make decisions about turning and driving based on the output of nbrGetNbrBearing(nbr). It will take no inputs and return True if your robot can see a guide robot with lower hops and False if it does not, along with the desired tv and rv for the robot.

2.1 Pseudocode
Provide proof of pseudocoding before you being programming and receive +5 points on the design challenge.

2.2 Scoring
+50 Robot makes it to the Source
- 5 For every guide robot your robot does not move towards