First, find the shortest path from H1 to P1, label it [A1, A2, A3... An], and the shortest path from H2 to P2, label it [B1, B2, B3... Bm]. Update these paths after each move with the same labels. Calculate the shortest path between A2 and B2(Predict the next move). Build a list of pairs to record the position of two dogs at each step. If the distance of the shortest path between the two dogs is greater than 1, continue.

If equals 1, there are three options. Always manipulate the longer shortest path. If two paths have equal distances, randomly choose one. Assume the path labeled A is the longer one:

1. Ignore A2, iterate through all adjacent nodes, and find the shortest path for this dog again. Label it [A’1, A’2… A’n]. If the number of nodes in this shortest path is the same as the number from A2 to An, and the shortest path from A’1 to B2 is greater than 1, choose this new path. If not, go to option two.
2. Wait at A1 for one step, let the other dog move to B2, then proceed. If the shortest path from A1 to B2 is smaller than 2, go to option 3.
3. Iterate through all adjacent nodes other than A2, proceed to this node if this adjacent node of A1 has a shortest path to B2 that has distance greater than 1. Then return to A1 as the other dog proceed from B2 to B3. If no adjacent node meets this condition, go through these 3 options for path labeled B.

If equals 0, there are two options. Always manipulate the longer shortest path. If two paths have equal distances, randomly choose one. Assume the path labeled A is the longer one:

1. Ignore A2, iterate through all adjacent nodes at A1, and find the shortest path for this dog again. Label it [A’1, A’2… A’n]. If the number of nodes in this shortest path is the same as the number from A2 to An, and the shortest path from A’1 to B2 is greater than 1, choose this new path. If not, go to option two.
2. Iterate through all adjacent nodes other than A2, proceed to this node if this adjacent node of A1 has a shortest path to B2 that has distance greater than 1. Then return to A1 as the other dog proceed from B2 to B3. If no adjacent node meets this condition, go through these 3 options for path labeled B.

Return the list of pairs.

Time complexity: O(n^3) There are 3 shortest path algorithms at each step, and there are n nodes in total. (3\*n^2)\*n. There could be new shortest path at each step, but this n^2 would not exceed the set O(n^3)