Question 1

1) Compute a strategy for controlling your dog bots that attempts to minimize the number of rounds needed to corner the sheep. How did you formulate this, and how did you solve it?

In this particular situation, we can model the problem like this: How to make the sheep dogs stand on the right side and the downside of the sheep? Based on the question premises, when the sheep dog is on the adjacent grid of the sheep, the grid which sheep dog is standing on will be blocked to the sheep. Thus, when the sheep dogs are standing on the right side and the downside adjacent to the sheep, it can only move to the left and upside. Thus, it will finally reach to the (0, 0) grid.

Thus, once we placed the sheep and dogs, we will randomly assign a destination to each dog, one of them will head to the right side of the sheep, another one will head to the downside. Then, we can formulate the distance between the destination(x_i, x_j) and the dog(y_i, y_j):

$$Manhattan\ distance = |x_i - y_i| + |x_j - y_j|$$

The aim of the chase process is to reduce of the Manhattan distance between the dog and the sheep. Then, we will go to the chase process. In this process, the two dogs will try to approach the sheep while the sheep will move randomly to its adjacent grids. When we are faced with the corner condition such as: 1. The sheep is at 3 corners of the map (0, 7), (7, 0), (7, 7). When it is at (0, 7) and (7, 0), two dogs will still chase the sheep, but one won't step on the final position it wants as the position is not valid. 2.When the sheep is on the border of the map. This situation is similar to the former condition, which means one of the destinations of the dogs is invalid.

2) Given the initial state in the above example, how many rounds do you need (on average) to corner the sheep? Answer as precisely as possible.

I ran 1000 times using the given points where sheep is in (4, 0), sheep dogs is in (0, 7), (6, 3). And the result is as below:

Average rounds	Failure(times)
19	0

The average rounds it takes is 19. The overall rounds of the 1000 unit runs is 19082. In this chasing algorithm, when one of the dogs has already reached the right side of the sheep and the other one(dog2) is right and downside of the sheep, we use a flip to avoid it from stuck. The function works like this: when we detect this situation, the dog2 will have 50% chance to jump in horizontal directions rather than the original choices (just in vertical direction). Applying this method can decrease the possibility of being stuck, but it will usually take more rounds for the dogs to corner the sheep. The table below will show the result of not applying this method (1000 unit runs):

Average rounds	Failure(times)
11	61

As you can see, the average rounds it takes decrease a lot in this method, but the failure has increased. Thus, this method of flipping is a trading off between success rate and corner speed.

3) What is the worst possible initial state (position of dog bots, sheep), and why? Justify mathematically.

The worst initial states of our algorithm are:

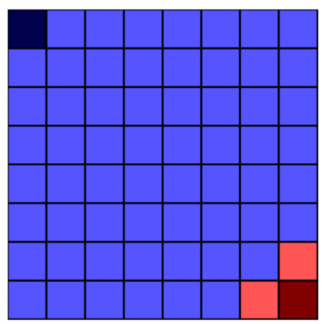

figure i sheep is cornered in the bottom right

The sheep is in (7, 7), and the dogs are in (7, 6) and (6, 7). Based on the algorithm we implemented, the sheep dogs will hold as the directions they choose will always invalid, thus, we can't solve this situation.

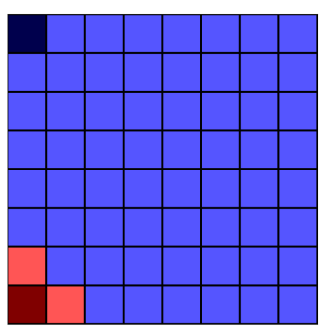

figure ii sheep is cornered in the bottom left

The sheep is in (7, 0) and the dogs are in (6, 0) and (7, 1). In this case, the dog in the upper side of the sheep would like to go to the (7, 0) while the sheep is already in there.

Thus, it will hold its position. The other dog will chase the sheep since it is already in the right position. But the sheep is already cornered, it will be stuck with the sheep. Based on that, the sheep will hold its place and can't move.

4) You are allowed to place your dog bots anywhere in the field at the start, then the sheep will be placed uniformly at random in one of the remaining unoccupied cells. Where should you place your dog bots initially, and why? Justify mathematically.

   In this problem, we can formulate it into the equation below, T represent the time:

$$T = T_{chasing} + T_{corner}$$

   We can divide the whole time it needs to two part: 1. The time dogs need to catch up the sheep. 2.After the dogs catching the sheep, time they need to corner the sheep. As the time we need for dogs to corner the sheep depends on the position where sheep is and the action it takes from the original place it stands. We can't optimize this part as the sheep will take random actions and we can't predict where it will go to. Thus, we will focus on how to catch up the sheep faster. As the Manhattan distance we discussed above and the probability of placing the sheep to each grid is the same. Thus, we would like to place the dogs in the middle of the map so the expectation of the distance will be the smallest.

$$E_{mahattan\ distance} = \sum_i \sum_j P_{ij} \times distance((i,j), dogs)$$

   P_{ij}represent the possibility of placing the sheep to grid (i, j). Thus, we would like to place the dog to (3, 4), (4, 3) or (3, 3), (4, 4).

   According to the formula we discussed above, we can optimize the chasing process of the whole rounds we need to corner the sheep.

5) Do you think better strategies exist than the one you came up with? Justify.

   Of course, yes. In this particular question, I think we can use Bayesian network to optimize our strategy. After observing the movement of the sheep, the dogs can get the information that the sheep may appears on which grid. Based on this information, the dogs can cooperate with each other and optimize the corner process. And we can avoid the worst situation happen as the dogs can predict the sheep's action. Thus, the overall performance will improve.

Bonus: Answer the same questions with the following modification: the dog bots win if they can corner the sheep in any of the four corners, but when the sheep moves it is allowed to strategize about which cell to move to, and wants to keep the dog bots from cornering it as long as possible.

1) In the modification, the basic strategy will be the same with the basic one. First, we need to catch up the sheep, then we need to corner it. In this situation, since the destination of the sheep is no longer limited to (0, 0), the dog can approach the dog in different directions. One of the solutions is similar with the basic one: 1. Detect the position of the sheep and assign the nearest corner as its destination. Then, we make dogs chase the sheep in the opposite two direction to corner the sheep. The reason why we want the dogs approach the sheep in two direction is that the sheep can strategize its movement to keep the dogs away. By approaching it in two directions we can minimize the rounds it takes corner the sheep.
2)
3) The worst position can be: the sheep is in (0, 1), two dogs are in (7, 7), (7, 0). In this situation, when the dogs are trying to approaching the sheep, it can avoid be captured in two directions. When it has three dimensions to move, it is really hard for it to be cornered.