Decision Making

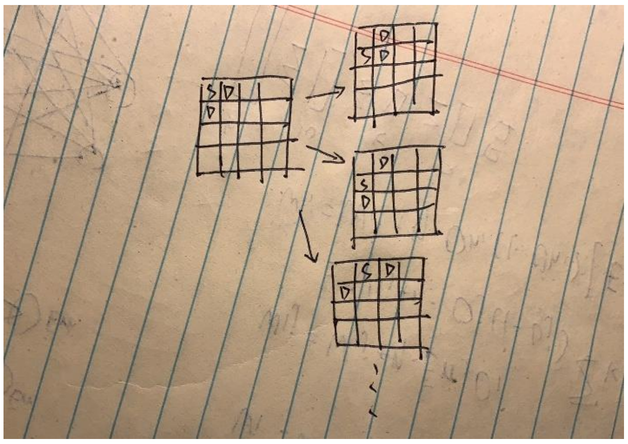
1) I've tried two strategies for this problem:

   The first strategy I use is telling the dogs where to go and let them find their own path.

   a) One dog should move to the bottom side of the sheep and the other moves to the right side.
   b) Apply A start algorithm, find path for each dog going to the bottom side or the right side of the sheep.
   c) There are two possibility in each move: Dog A heads to the bottom side while dog B heads to the right side and Dog A heads to the right side while dog B heads to the bottom side. We compute the cost of the two situation and see which one can shorten the distance between two dogs and the sheep more.
   d) We pick that strategy from the previous step and continue making the decision until the sheep got to the destination.
   e) Notice that if the sheep is at the last row or column, its bottom side and right side is no longer valid, if this happens, the dogs should do to the left/right and bottom/top side and wait until the position is valid again.

   For the second strategy, I am trying to solve this problem with by applying searching theory.

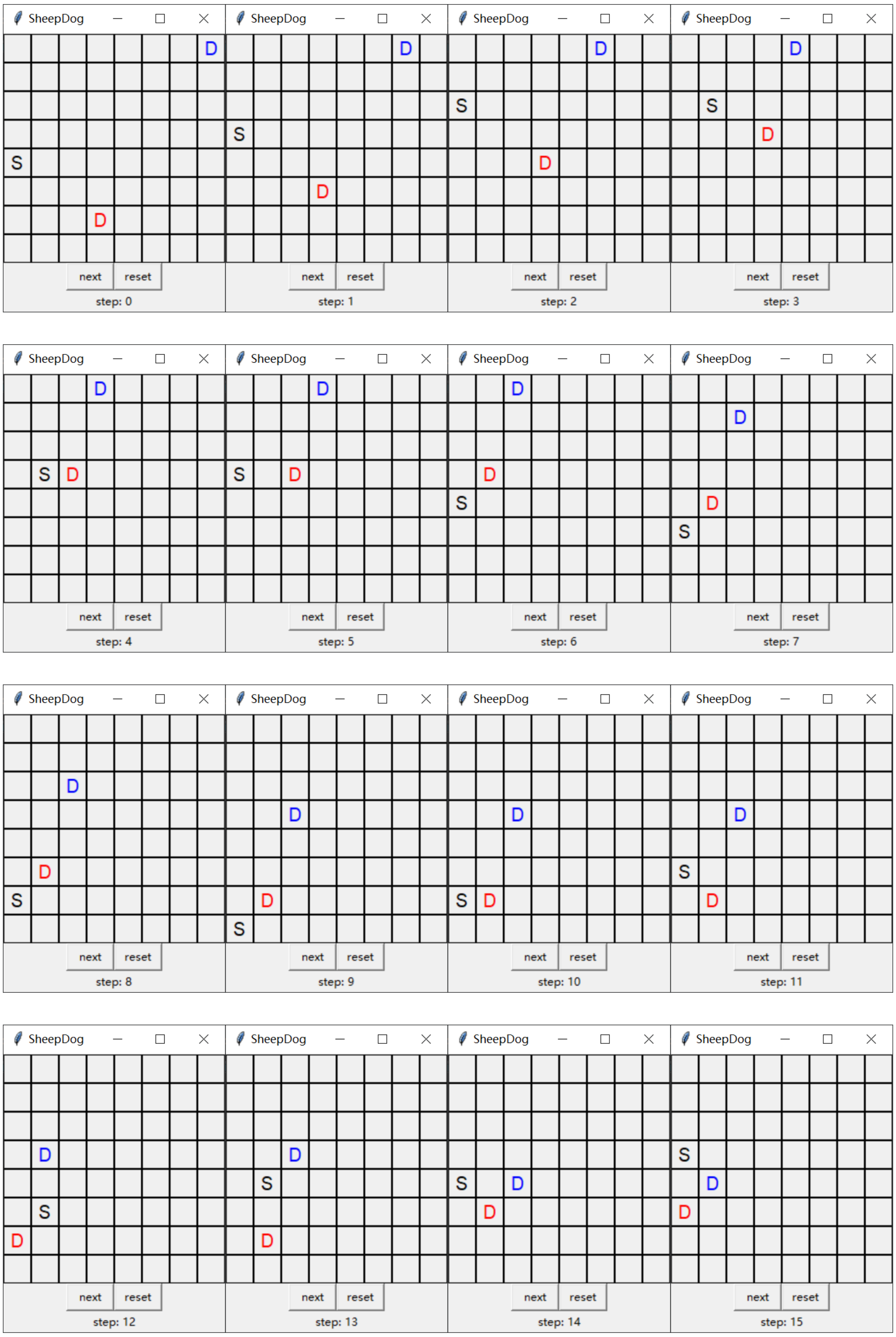
   a) We assume that the final state of the model always ends at sheep being in the position (0, 0), we push all such final state in a list.
   b) We traverse the list and for each state, we explore all its children states by moving the position of sheep and sheepdog, we record their relationship with their parents and stored the number of steps we take to reach that state. We push all these states into the list.

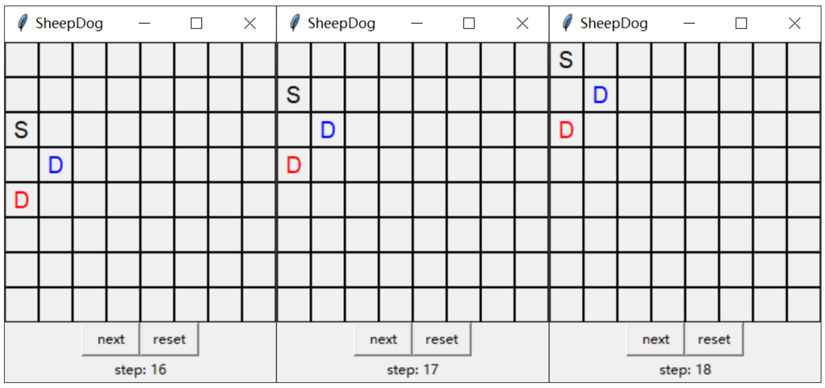


   c) While iterating, if we find that the number of steps can be reduced by changing the parents, we change its parents and update new number of steps.
   d) After the list is empty, the path is generated. Started from a state and keep tracking its parents until we got to the final states.

2) I run my algorithm and pick a bad case where the sheep heads to the bottom at the beginning, and here is the performance of my algorithm:

First strategy:

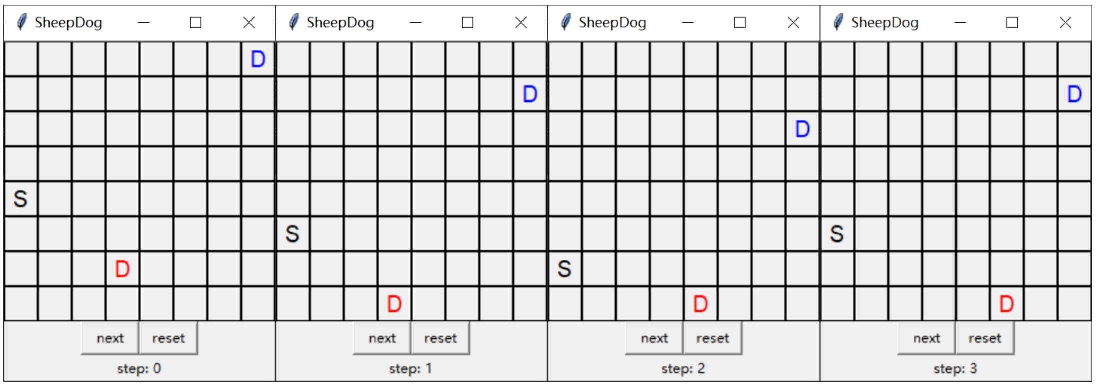


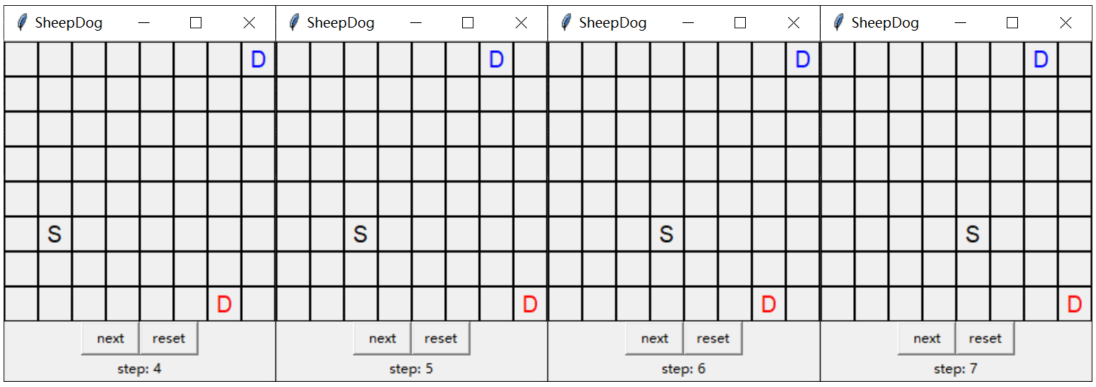


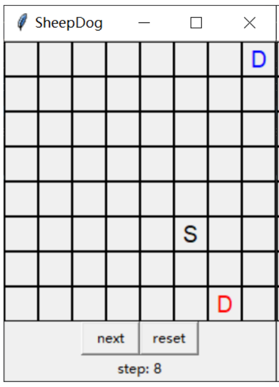
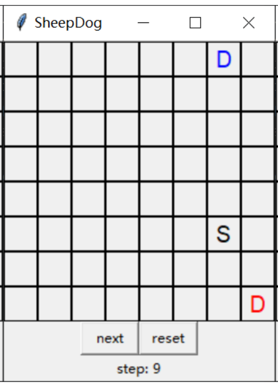
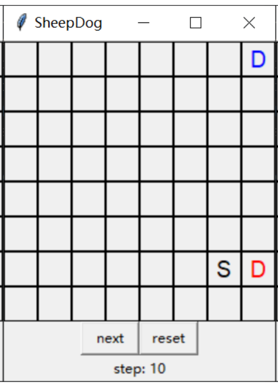
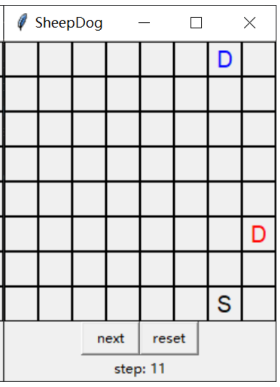
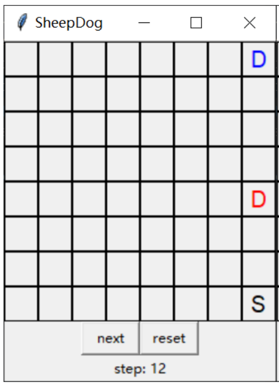
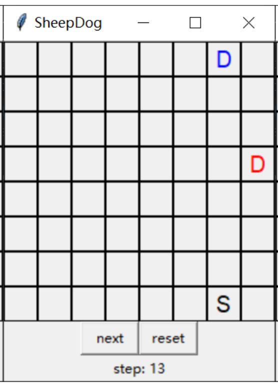
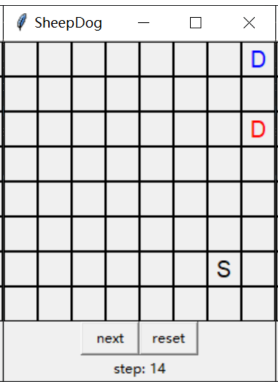
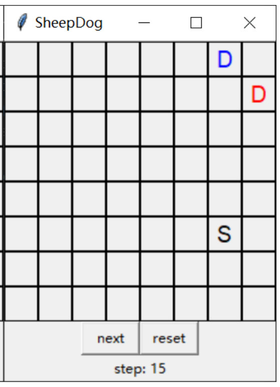
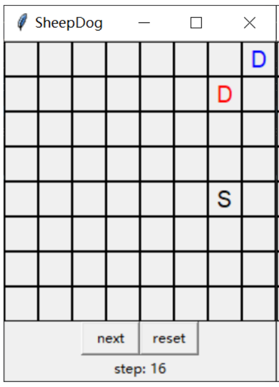
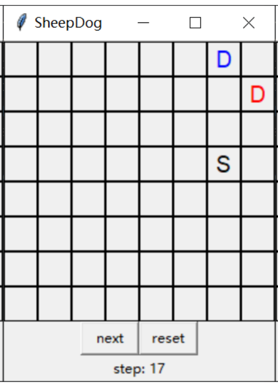
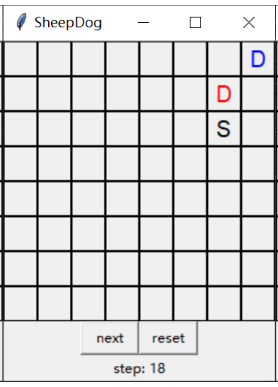
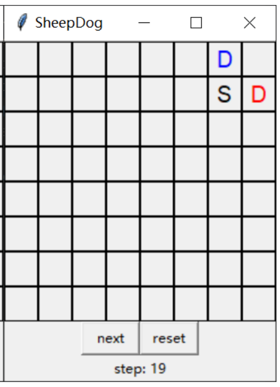
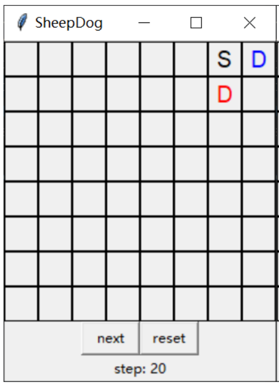
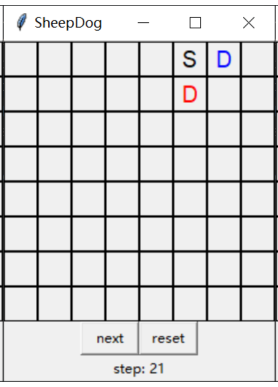
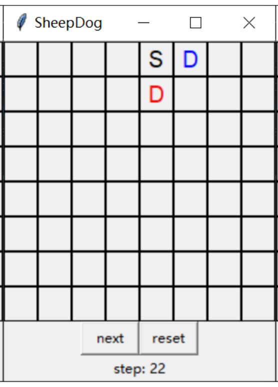
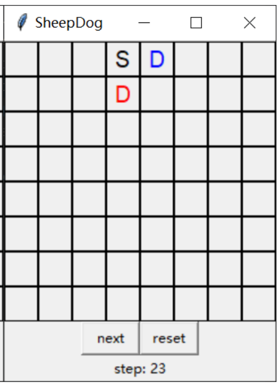
Notice that the sheep takes action after dogs, so the last serval steps do not mean that the sheep doesn't choose to go right but actually the dogs block its way of going right instantly.

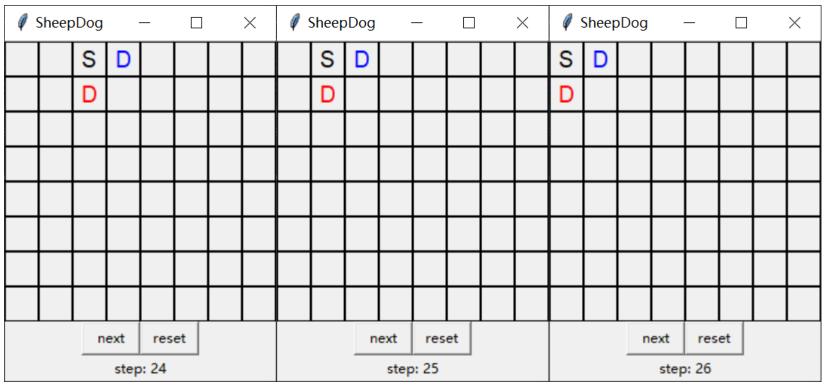
On average, the steps needed is 14.

Second strategy:



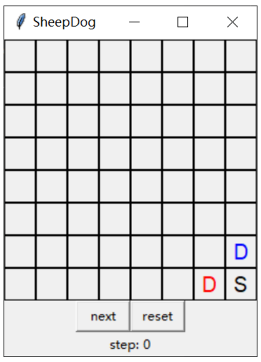



SheepDog
D
S
D
next
reset
step: 8

SheepDog
D
S
D
next
reset
step: 9

SheepDog
D
S
D
next
reset
step: 10

SheepDog
D
D
S
next
reset
step: 11

SheepDog
D
D
S
next
reset
step: 12

SheepDog
D
D
S
next
reset
step: 13

SheepDog
D
D
S
next
reset
step: 14

SheepDog
D
D
S
next
reset
step: 15

SheepDog
D
D
S
next
reset
step: 16

SheepDog
D
D
S
next
reset
step: 17

SheepDog
D
D
S
next
reset
step: 18

SheepDog
D
S
D
next
reset
step: 19

SheepDog
S
D
D
next
reset
step: 20

SheepDog
S
D
D
next
reset
step: 21

SheepDog
S
D
D
next
reset
step: 22

SheepDog
S
D
D
next
reset
step: 23



The behavior is kind of weird and it takes more steps than the first strategy. I guess it is because as the sheep is moving randomly in the board, the algorithm fail to react towards the behavior of the sheep.

3) For the first strategy, the worse initial state is shown as following:



In this case, the bottom side and right side of the sheep is invalid, and the sheepdogs need to go backward to make a way for the sheep.

For the second strategy, the previous case would also be the worse initial state as it has largest Manhattan distance to the final states.

4) We would need to minimize the distance between the dogs and the sheep and the distance between the sheep and the final position:

$$\min\,({s_1}^2 + {s_2}^2 + (s_1 - d_1)^2 + (s_2 - d_2)^2\,)$$

As the location of the sheep is picked randomly, the dogs should be placed near the center of the board.

5) There is a better strategy than my first strategy as for the first strategy the dogs are just heading to the positions I told them. They are not going to think of the situation and take reaction themselves.

There is also a better strategy compared to the second strategy as the second strategy is basically storing all $64 \times 64 \times 64$ possible situations and making connection between them. However, as the sheep is moving randomly, it cannot guarantee that the things happens as we predict.