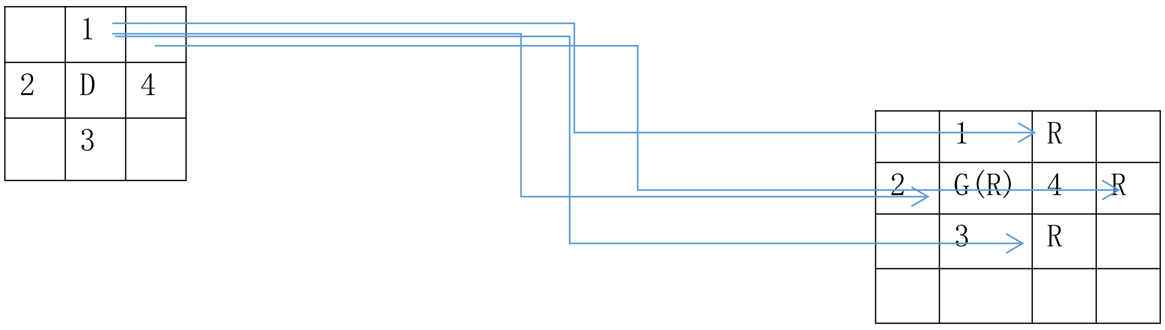
Ruolin Qu
CS520 Intro To AI
12/20/2019
Report of question 1

1.
For this question, we are required to implement a optimal strategy rather than a “fancy” algorithm we come up with to solve this type of question. The algorithm I utilize, borrowed the idea from MDP(Markov decision process). For each round, our knowledge base contains basically these elements: Action(for both goat and dogs), State(location), Transition function(from one state to another state).

Thus, the first I consider is the reward from the current state to the next state. In this 8*8 maps, we have two dogs, our job is to move those bogs by pre-set rules and surround the random moving goat in the left upper corner. In this case , I first calculate which dog is good for the right position, which dog is good for the bottom position in the surround circle. Next step is the most important, define the next available actions. For this steps , I need to come up with a reasonable evaluation for all actions, in other word, setting reward function. The most efficient strategy I tested is in the following: for each action right chaser dog, I calculate the Manhattan distance from the possible right location for goat and add them up.



Just like the mapping relationship above, then for these four possible dog actions, we choose the one with the smallest the smallest sum of Manhattan distance as the highest future reward.

2.

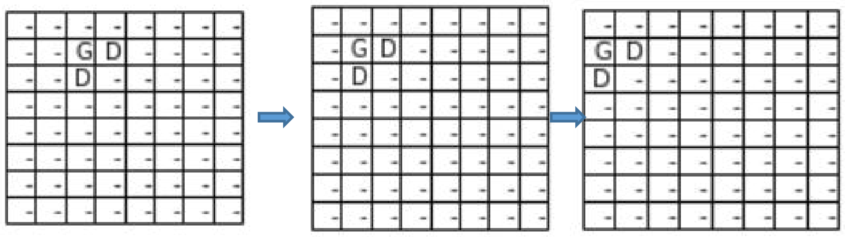
For this test, I calculate the average rounds need to take to get to corner for 100 times, the average is 92.48. It also showed in the program.

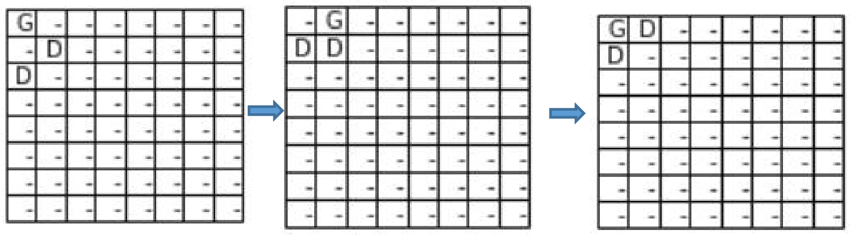
3.
The worst state should be the initial position of dogs are both in the upper left and the goat is in the bottom right. To justify, the goat is randomly moving in the map. Assume we do not have dogs in the maps, the final position of goat will uniformly distribute in the center of its initial position. In this circumstances, our dog have to travel all the way down to start effect the movement of goat. Considering we set the reward function appropriately and two dogs will not influence each other, both of them travel the N cells horizontally and N cells vertically,which cost O((Length+Width)*2) time complexity, which is the highest for all the situations:

$$TotalCost = 2*(length + width) + \lambda chase\_dis\tan ce$$

λ is the factor associated with our algorithm, the better reward and transition function we set up, the smaller it will be.

4.
I will put one dog at next right cell and one dog at bottom. In this way, the dog will lead the goat to the upper left corner efficiently from the beginning. Just like the screenshot accordingly:





The reason is similar like we mentioned above,the total cost for each iteration is decided by two fators: first, the distance between two dogs and the goat. Second,the efficiency dogs lead the goat to the destination, rely on the algorithm we design:

$$TotalCost = 2*(length + width) + \lambda chase\_dis\tan ce$$

Given the condition of fix location of the goat, the chase_distance is a constant, we can arrange the dog next to the goat to minimize the length and width distance needed to reach and have a early impact to the moving direction of goat.

5.
There should be better strategy:
First, based on my design, we have a dog is responsible for driving the goat to the left and the other is responsible for driving goat toward upwards. However, how to decide which one is suitable for those job is a way to improve my algorithm. For instance, right now I decide the rightmost one to take the first job, there should be better assessment worth to take.
Second, the reward function can be improved, so far I take four future position and its current state as consideration. However, I notice it always worth to try to shrink the effect of the future position as the progress goes, the specific shrink strength can be added in this stage.