Assignment 1

CHAPTER 2

Ex. 2.3

(a) An agent that senses only partial information about the state cannot be perfectly rational.
False: An autonomous car can have objects (vehicles, stop-signs, cyclists, pedestrians) beyond its camera's scope, and thus the universe can be only partially observable. In spite of not being able to see everything, the car can still make rational decisions.

(b) There exist task environments in which no pure reflex agent can behave rationally.
True: Because of a pure reflex agent's inability to remember percept history, its actions will not always be rational. Take for example the vacuum cleaner world (with just 2 locations). The cleaner keeps oscillating back-and-forth if both the places are always clean.

(c) There exists a task environment in which every agent is rational.
True: Such a task environment will be extremely simple. Say the vacuum cleaner world has just one location. The rules are that the cleaner sucks dirt if dirt is present else does nothing.

(d) The input to an agent program is the same as the input to the agent function.
False: The input to an agent program is just a percept whereas the input to the agent function is a sequence of percepts.

(e) Every agent function is implementable by some program/machine combination.
False: Implementation is possible only if the program/machine combination has the necessary hardware and software resources for the computation. Also, if the problem in hand is theoretically intractable, the agent function can never be implemented.

(f) Supposed an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.
True: If all possible actions are rewarded equally, an agent selecting an action randomly from the set of possible actions can be deemed rational.

(g) It is possible for a given agent to be perfectly rational in two distinct task environments.
True: If an agent has a clear idea of what actions make sense in which environment, it can be perfectly rational in the two (or even more) distinct task environments.

(h) Every agent is rational in an unobservable environment.
False: It's very plausible and obvious that in an unobservable environment, there might be agents whose actions have no logical meaning.

(i) A perfectly rational poker player agent never loses.
False: However rational a poker player agent is, it never knows the cards its opponent might have. Since the universe is only partially observable, the agent might still lose even after making rational decisions based on the percepts.

Ex. 2.9

Please refer to `Question 2 python code.py` and `Python code output.doc`

CHAPTER 3

Ex. 3.3

(a)
Say friend A lives in city A_1, while friend B lives in B_1. The cities visited by A are A_1, A_2, A_3, A_4, …., while B visits B_1, B_2, B_3, B_4, and so on.
At every turn, the time taken = $\max\left(d(A_n, A_{n+1}), d(B_n, B_{n+1})\right)$
Goal: $A_n = B_n$ as quickly as possible.

(b)
Admissible heuristic functions do not overestimate the actual distance between any node and the goal node. Therefore (i) $D(i,j)$ and (iii) $D(i,j)/2$ are admissible while (ii) $2 * D(i,j)$ is not.

(c)
If the two friends end up in 2 neighboring cities, they'll keep swapping places unless there's another alternative route along which there are an odd number of cities. In fact, from the very beginning, if there's no path on the entire map where the route taken by the friends has an odd number of cities, the map will not have any solution.

(d)
If the map is such that any possible path between the 2 friends will certainly have an even number of cities, then at one point one of the friends will have to "revisit" a city, in that he will not change cities while his friend progresses to his next city. "Revisiting" a city is the only way by which such a map will have a solution.

Ex. 3.4

To show that all possible states of the 8-puzzle game can be divided into two disjoint sets, the numbers should be written down linearly, excluding the blank tile. For eg.:

3	2	
5	4	1
6	8	7

Writing down the numbers linearly: 3 2 5 4 1 6 8 7
Starting from the left, for every number ***N***, count the number of numbers to its right that are less than ***N***, add them up.

3	2	5	4	1	6	8	7

For ***N*** = 3, there are 2 numbers.

3	2	5	4	1	6	8	7

For ***N*** = 2, there is just 1 number.

3	2	5	4	1	6	8	7

For ***N*** = 5, there are 2 numbers.

Continuing, for ***N*** = 4, there's 1 number; for ***N*** = 1, it's 0; for ***N*** = 6, it's 0; for ***N*** = 8, it's 1; and for ***N*** = 7, it's 0 as well.

Summing them, it's 2 + 1 + 2 + 1 + 0 + 0 + 1 + 0 = 7, which is an odd number. Now it so happens that irrespective of how the blank tile is moved around for this particular configuration, writing down the numbers linearly and performing the same operation as above will always yield an odd number.

Furthermore, of all possible permutations of the 8-puzzle game, half of them result in even numbers and the other half in odd numbers. It is impossible to transition from one state to another such that the current state produces an odd number while the previous state was an even numbered state. Hence, the even and odd numbered states form two <u>disjoint</u> sets, and no state can be reached from any state in the other set.

The procedure explained above can be used to decide which set a state belongs to, and it's also worthwhile to note that the goal state must be in the same set too.