CS 4365 Artificial Intelligence: Project 1
Sean Summerfelt

The environment is Fully Observable, as the agent knows the entire state, Single Agent, as only the traffic light is an agent, Deterministic, cars will always act exactly the same way, Static, because the environment can't change during computation, Discrete, because actions happen in discrete time units and Known, because all of the rules of the simulation are known.

For this task, I decided to consider only actions with do not occur in a collision, in this case when no cars are going, when 1 car is going straight, when 1 car is going to the left, and when 1 car is going straight and the car to it's left is going left, controlling which lanes can go by turning on and off the red light for both the main array and the arrow. I choose a greedy best-first algorithm with the heuristic that picks the next node where the maximum number of cars are going through the intersection, choosing the lanes with the longest queues for tiebreaking. While this strategy isn't optimal, it is easy to implement, reasonably close to optimal, although I don't know how close, is O(N), the best complexity that can be reasonably expected from this problem, and it is complete: it can't fail because of the problem formulation, will always get closer to the goal, and can't get stuck.

The implementation is designed to be close to the description of the problem. The cars are put into queues when they arrive, and are implemented as objects, even though for the purposes of the algorithm the queue could simply be represented as it's length. If I could change anything about the design, it would be to add some code to change the pattern of light states into something that more closely approximates actual traffic; as it is, the code controls traffic purely by turning on and off red lights, and never touches green or yellow.