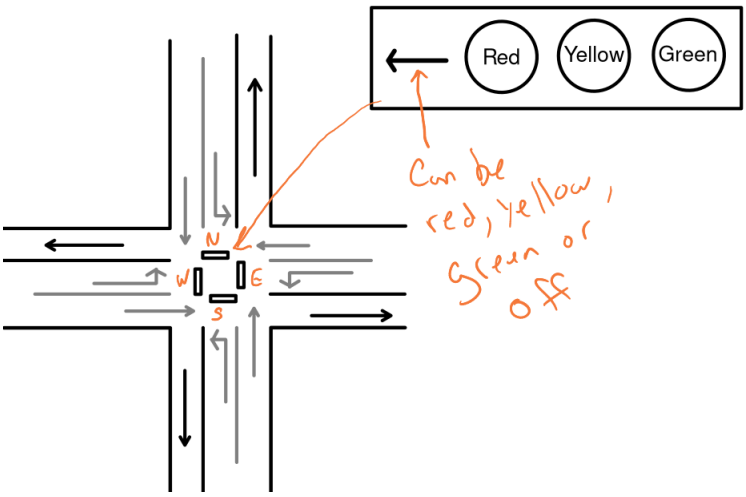
Problem Solving Through Search

Due: Wednesday, Feburary 26th

1 Description

Imagine there is an intersection, where two roads cross. Each road has two directions, and each incoming direction has two lanes. The left lane is reserved for those that want to turn left, and the other lane for those moving straight. Facing each direction is a traffic light, with a red light, yellow light, green light, and an arrow. The arrow can either be red, yellow, green, or off, and all other lights can either be off or on.



Each lane can be viewed as a FIFO queue, with the first car being "at the intersection" and all other cars "waiting for the intersection". For this project we will be using discrete time so it takes a single time step to cross the intersection and be replace with the next car.

A car that is at the intersection can go straight only if they are in the right lane and the red light is not on. However, if a car is at the intersection the time unit after the green light is *turned on* it will go into the intersection regardless of the condition of the red light.

When a car goes straight it's destination is the road it started from. If a car turns left its destination is the road it did not start from. Two cars collide if they enter the intersection during the same time step, but have different destinations.

To keep things simple, we will assume that when a car enters a road it will wait in the desired lane. So you do not have to worry about the case where a car, wanting to turn left cannot because it cannot get into the lane. In addition, we will assume that you as the designer know exactly what cars will arrive at the intersection, at which times, and their desired lane for the day ahead of time.

The goal of the agent is to plan out a schedule for the lights so that all cars can go through the intersection and no cars collide.

Some other things you may want to consider: How long does a car have to wait on average? Are there times cars can go through an intersection when there are no cars

available? Is there any strange behavior that does not affect the goal, such as the yellow light always being on? etc.

2 Input

We will be assuming that the agent will be provided with the exact behavior of the cars for the day. This input will be given in the form of a text file, with each line specifying the arrival of a car in the following format:

```
TIME ORIGIN DIRECTION
```

The time parameter specifies the time step the car will arrive at the intersection. If there is already another car at the intersection, then this car will wait for the intersection. The origin parameter specifies which traffic light the car will watch (and there for which road and direction it comes to the intersection from). It will be a single letter either N,S,E, or W. The direction parameter specifies whether or not the car will turn left. The parameter will either be left or straight. So, for example:

```
3 S left
```

This specifies that a car will arrive at the south part of the intersection at time step 3, and with the intent to turn left. To keep things simple you may assume that the lines in the file are ordered so that the time step is in ascending order.

3 What to do

You will design the Formulate/search portion of an agent for the traffic lights. Your agent will be given as input the traffic information for the day, and must plan out the schedule for the lights.

- Provide the Task environment for this task
- Formulate the Problem
  - Define all five pieces of information
  - You may determine anything not specified in this document. For instance, the step cost is not defined.
- Choose a search strategy
- implement the search strategy to compute the solution for any given input
  - print the schedule to standard out as a sequence of time important time steps(then changes occur).
    - The time stamp is simply the time step the even should happen
    - For each traffic light (labeled N,S,E,W):
      - For each light in order of red, yellow, green output either on or off
      - After the three color lights, for the arrow light out put either, red, yellow, green, or off.
- Write a write-up as specified in section 4.

4 The Write-up

You will need to turn in a write-up along with your source code. The write up should include:

- The task environment
  - List the properties, as presented in chapter 2, of this task environment with a short explanation (one or two sentences each are fine).
- Problem formulation with an explanation of the choices you made
- The answers to the following questions:
  - What search strategy did you choose?
  - Why did you choose that particular search strategy?
  - What other design decisions did you make? Why did you make the choices that you did?
  - Is your agent complete? Is it optimal? Why?
  - If you could change one thing about your design, what would it be?

5 What to Turn in

Upload your submission as a zip archive containing the following:

- Source code (c, c++, python, or java files)
  - Source code should not require a particular IDE to compile and run.
  - Should work on the cs1 and cs2 machines
- Readme (Plain text document)
  - List the files included in the archive and their purpose
  - Explain how to compile and run your project
  - Include any other notes that the TA may need
- Write-up (Microsoft Word or pdf format)
  - As specified in section 4
  - In addition, if you did not complete some feature of the project, why not?
    - What unsolvable problems did you encounter?
    - How did you try to solve the problems?
    - Where do you think the solution might lay?
      - What would you do to try and solve the problem if you had more time?

6 Grading

The grade for this project will be out of 100, and broken down as follows:

Followed Specifications	50
Correct Output	10
Write-up (Including task environment and problem formulation)	40

If you were not able to complete some part of the program discussing the problem and potential solutions in the write-up will reduce the points deducted for it. For example, suppose there is a bug in your code that sometimes allows two customers to approach the same worker, and could not figure out the problem before the due date. You can write 2-3 paragraphs in the write-up to discuss this issue. Identify the error and discuss what you have done to try to fix it/find the problem point, and discuss how you would proceed if you had more time. Overall, inform me and the TA that you know the problem exists and you seriously spend time trying to fix the problem. Normally you may lose 5 points (since it is a rare error) but with the write-up you only lose 2. These points can make a large difference if the problem is affecting a larger portion of the program.