CSCI 3202 - Introduction to Artificial Intelligence
Instructor: Hoenigman
Assignment 3
Due Friday, September 16 by 4pm

Problems:

A* Search
There is a .txt file on Moodle that contains a list of nodes and edges that can be used
to build a graph. The first few lines of the file look like:

```
[S, A, 5]
[S, B, 1]
[S, C, 4]
[B, E, 9]
```

You can read this as, there is a node called S, which connects to nodes A, B, and C
with edge weights of 5, 1, and 4, respectively. The node B connects to node E with an
edge weight of 9.

The .txt file also includes heuristic values for a search from S to F. For example,

```
A=5
B=6
```

means that the estimate for the cost from A to F is 5, and the estimate for the cost
from B to F is 6.

Write a program to do the following

1. Build a graph from the text file provided.
2. Apply A* search and Dijkstra's algorithm to find the shortest path through
   the graph. The path includes the cost and the nodes traversed.
3. Compare A* and Dijkstra's for how long it took each algorithm to find the
   solution. The comparison needs to include the number of nodes evaluated.

Document your results
   - In addition to the code you submit, you need to submit a write-up that
     describes your results.
   - Included in the write-up:
       o Purpose of the assignment.
       o Data used: Include a picture of the graph generated from the .txt file.
       o Procedure: The algorithms you implemented and ran
       o Results: The shortest path that each algorithm generated and the
         nodes that each algorithm evaluated as the algorithm was executing.
         Comment on whether either algorithm was more efficient, i.e.

evaluated fewer nodes. If they were the same, provide an example where a modification to the data would have provided a different result.

What to submit
Zip your source code and write-up and submit it to moodle. In assignment 1, we wrote a simple graph class. You are welcome to use that code as your starting point for this assignment. You will need to modify the nodes to include additional properties to support A* and Dijkstra's algorithm.

Other details
The TAs will assume your code is in Python 2 unless it is really clearly stated that you used Python 3. They will grade your code by running it command line in Ubuntu 16.04. The name of the file used to build the graph needs to be a command-line argument to your program.