Paul Laliberte'
CSCI-3202, Artificial Intelligence
Dr. Hoenigman
Assignment 3

Purpose of Assignment

The purpose of this assignment was to create our own graph, given a text file with nodes, edge weights, and heuristics. We then implemented Dijkstra algorithm and A* search. Comparing these two types of search algorithms results in identifying that Dijkstra, even though it was developed first, is very similar to A* search. In fact, Dijkstra's algorithm is A* search with heuristic values set to zero, or rather the heuristic function

$$f(n) = g(n) + h(n),$$

where $h(n) = 0$. This gives us the function for Dijkstra's algorithm, $f(n) = g(n)$.

Data Used

The text file provided was used in implementing the graph:

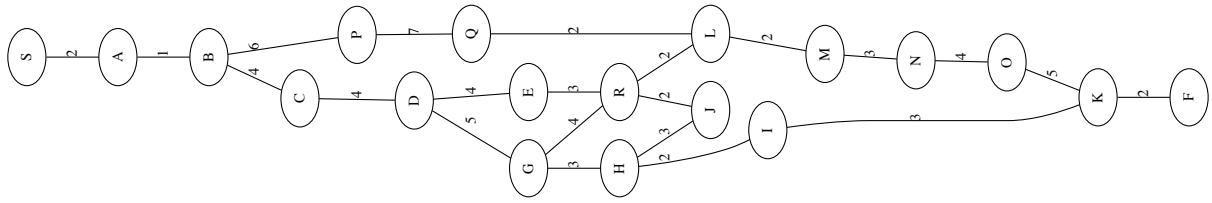

Figure 1: Graph

Procedure

I first implemented a graph class and a vertex class. Then I modified my make file to open and read through the assignment3.txt file, and properly added them into the graph. Once I had a connected graph with the proper weighting between edge nodes, I implemented Dijkstra's algorithm using a queue and the python import library "heapq". I was able to reuse most of my code from Dijkstra's algorithm for A* search. In A* search I did implement a separate heuristics function that would get the heuristics for a node when called upon. The proper addition to the cost function $f(n)$ was also added to A* search.

Results

Given the input file as is the shortest path was the same for both Dijkstra's algorithm and A* Search: S,A,B,C,P,D,Q,E,G,L,R,M,H,I,N,K,F. Both algorithms were equally efficient in this manner, under the assumption that we are only considering nodes solved and shortest path (excluding time complexity, space, etc.). A modification to the C-heuristic would have changed the A* search to dive down the "right" path (top path on graph figure above). I implemented this change to the assignment3.txt file, and the result of the change is output along with the original search when the make file is ran.