**Assignment 3: THE HOBBIT REUNION**

**How to Compile:**

1. Once assign3.c is downloaded to the designated folder, open your terminal.
2. Go to the directory in which assign3.c are located.
3. Type in the terminal:
   1. gcc -o assign3 assign3.c
4. assign3.c are now compiled.
5. Type-in ./assign3 to run the program.

**How to Use and Summary of Observations:**

The client can put 4 commands: SHP, SDP, STP, FTP

1. **Shortest Hop Path (SHP):** This algorithm finds the shortest path from source to destination, where the length of a path refers to the number of hops (i.e., links) traversed. Note that this algorithm ignores the physical distance, travel time, gold coins, and trolls for each link.
2. **Shortest Distance Path (SDP):** This algorithm finds the shortest path from source to destination, where the length of a path refers to the cumulative total distance traveled. Note that this algorithm ignores the number of hops, travel time, as well as gold and trolls.
3. **Shortest Time Path (STP):** This algorithm finds the shortest path from source to destination, where the length of a path refers to the cumulative travel time for traversing the chosen links in the path. Note that this algorithm ignores the number of hops, as well as the distance (although distance and time are often correlated). Gold and trolls are also irrelevant.
4. **Fewest Trolls Path (FTP):** This algorithm finds the path from source to destination that minimizes the number of trolls encountered. Note that this algorithm ignores the number of hops, as well as time, distance, and gold (although gold and trolls are often correlated).

In this program, I have used dijkstra’s algorithm in which it finds the solution for a single source shortest path problem. For all algorithms, whenever ties occur, the first path found is chosen. For any network topology, this is stored in 5 2D arrays for each category (e.g., distance, time, gold, troll, and hops). Once these are stored, then for each routing algorithm uses the relative 2D array to find the cost of the traversal. Each routing algorithm calls the dijkstra method and stores its values to the cost arrays and depending on the path traversed, the values of the remaining categories specified are also calculated. Once, all the cost values for all the categories are calculated, this then calculates the averages for each category and prints out to the screen terminal the routing result table.

Note: I have used Virtual Box running Ubuntu in creating this algorithm.