Daniel Gregory

Project 3 Writeup

To get the program to run I placed the data files inside the Project 3 folder so it can find it easily. The first thing this program does is open the two text files and save the vertices then the edges and constructs the graph of middle earth. The next helper function is for implementing a priority queue using a binary heap which is later used in Dijkstra’s. This helper function though has different parts that help move elements up or down as well as to insert elements and extract the minimum element from the heap. The final part of this is there are helper functions that help you get the parent, left, or right child indices. The Breadth First Search is the next function here and it is fairly straightforward as it just progresses through the tree finding the shortest path between two points and prints it out. While Dijkstra’s used the binary heap that was created at the start in order to extract the shortest distance from one point to another and prints the map out. The final algorithm is Prim’s algorithm and I had ChatGPT help to try and figure out how to do it. I just asked it how and it gave me steps to follow to get it to work. Then just had to follow them and update the code so all of the parameters were correct and then formatted in the right way and it worked. The main function here is short and just calls all the functions listed above as well as some lines of code to help with the formatting of all of this.

Shortest path from Hobbiton to MountDoom using BFS:

Hobbiton -> Southfarthing -> Isengard -> Edoras -> Rauros -> BlackGate -> MountDoom

Shortest Path from Hobbiton to MountDoom using Dijkstra's:

Hobbiton -> Bree -> Rivendell -> Moria -> Lorien -> Rauros -> BlackGate -> CirithUngol -> MountDoom