Matthew Aguilar

Program 3 Design Document

To Compile:

In order to compile my program, one must just run the make command, which will build the dijkstra executable

To Run:

In order for my program to work, one must execute it as follows:

    ./dijkstras [valid input text file name]

In order for an input text file to be valid, it must be in a specific format:

    [N = number of nodes in graph]
    [names of nodes separated by spaces]
    [NxN Adjacency matrix for the nodes, with 9999 representing infinity, or the absence of edge between two nodes]

My program will print the adjacency matrix and then wait for input from the user, which must be the name of an existing node in the graph.
Finally, my program will print the result table as it calculates the shortest path from the starting node provided by the user to all other nodes in the graph.

Design Decisions:

I used ifstream to read lines from the input files, and then I use istringstream to extract data from each line of the input file.

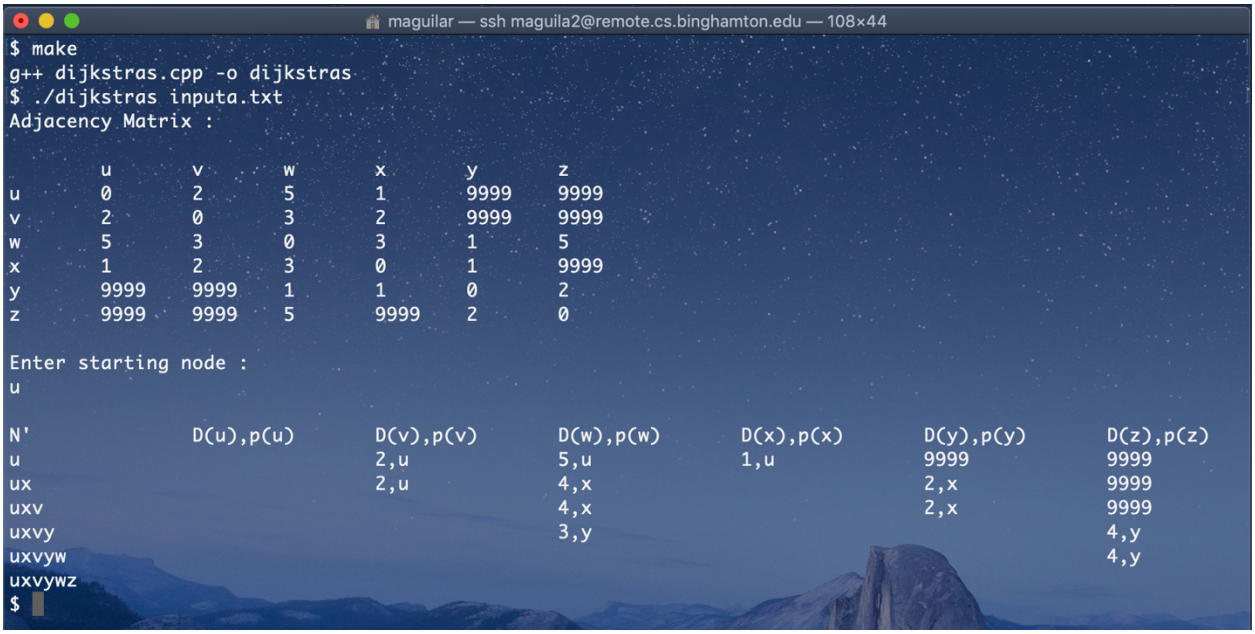
Once I get the number of nodes and their names, I set up the adjacency matrix so I can then print it. I decided to make the adjacency matrix a 2D int array in order to conserve memory. Then I get user input to select the starting node for which the algorithm will determine the shortest path.

Before calculating the shortest path though, I make a few vectors to store different data for me about each node (like whether it has been visited, the distance from the node to the starting node, and the previous node in the shortest path).

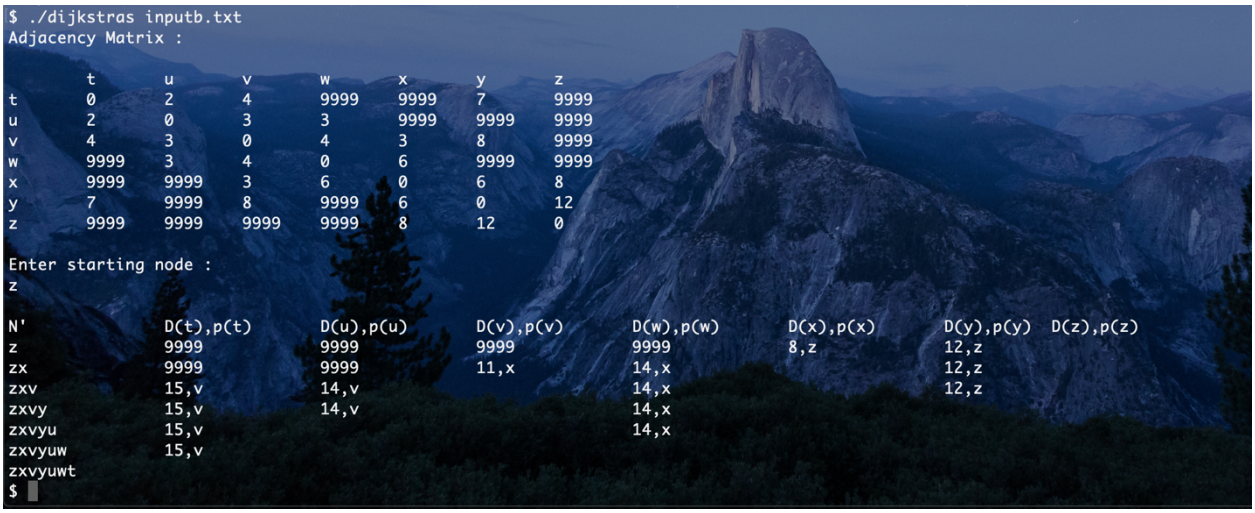
The main loop of Dijkstra's algorithm finds an unvisited node that has the shortest distance from u and uses this node to update the shortest distance from u to all other nodes. After each run, I print out the updated row of the result table.

I had to make another vector of nodes just for the visited nodes (named nPrime) so I could easily print out the nodes that are in N' in the order that they are added.

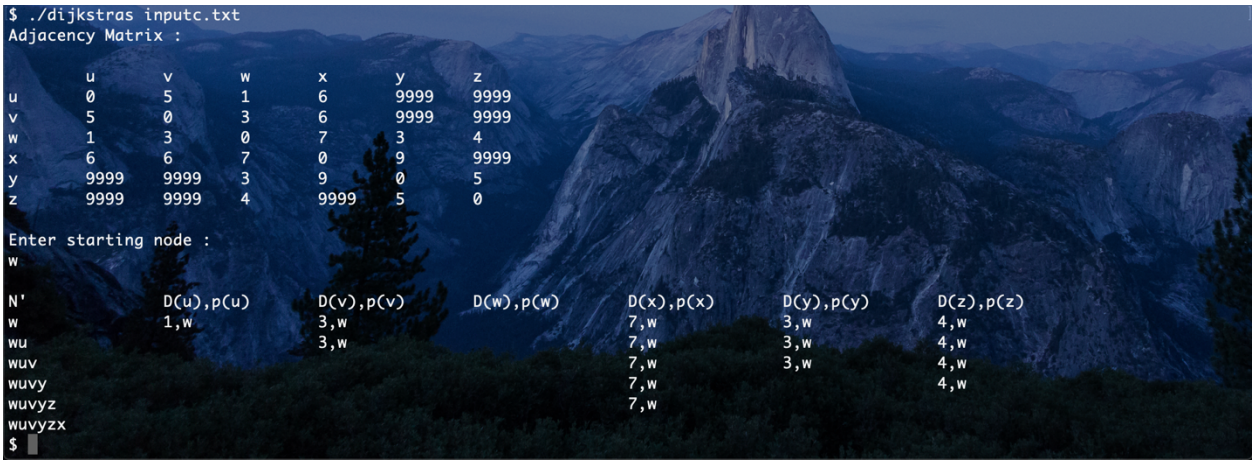
inputa.txt



inputb.txt



inputc.txt



```
$ ./dijkstras inputc.txt
Adjacency Matrix :

        u       v       w       x       y       z
u       0       5       1       6       9999    9999
v       5       0       3       6       9999    9999
w       1       3       0       7       3       4
x       6       6       7       0       9       9999
y       9999    9999    3       9       0       5
z       9999    9999    4       9999    5       0

Enter starting node :
w

N'              D(u),p(u)       D(v),p(v)       D(w),p(w)       D(x),p(x)       D(y),p(y)       D(z),p(z)
w               1,w             3,w                             7,w             3,w             4,w
wu                              3,w                             7,w             3,w             4,w
wuv                                                             7,w             3,w             4,w
wuvy                                                            7,w                             4,w
wuvyz                                                           7,w
wuvyzx
$
```