CSCI 630

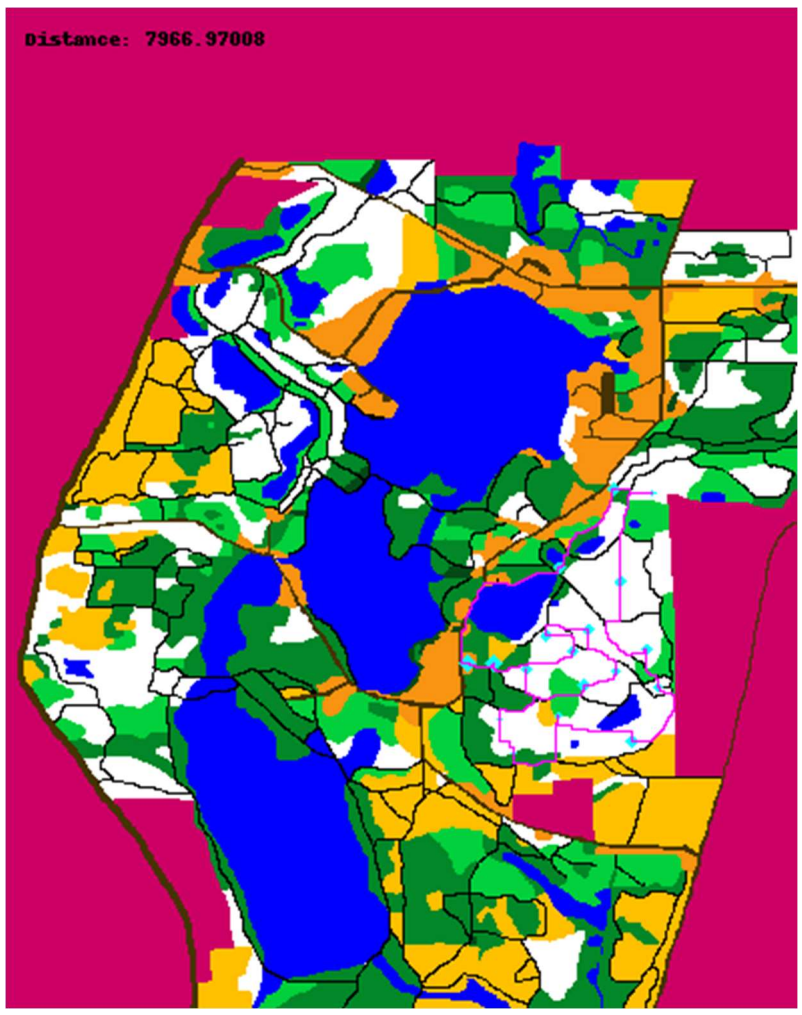
Lab#1

Aditya Ajit Tirakannavar – at2650@rit.edu

Execution command:

python lab1.py terrain.png mpp.txt red.txt redOut.png

output:




This program is solved using A* Algorithm and BFS algorithm.

A* is a combination of Dijkstra and Greedy Algorithm. It uses the distance from the root node plus heuristics distance to the goal. The algorithm terminates when we find the goal node.

A* formula:

F( x) = G(x) + H(x)

where F( x) = totalcost

Gx = cost from current pixel to neighbour pixel

Hx = heuristic function (straight line distance to target)

BFS is used to find the optimal coordinate points through which paths are drawn

Libraries:

I've used 5 python libraries – **PIL**, **sys, re, heapq** and **math**.

- PIL is used to work with image file.
- The split function of the Re library is used to split the elevation values while reading from the file as the spaces are uneven
- sys is used to take arguments from the user
- math is used to perform arithmetic operations for calculating the distance.
- Heapq is used in order to implement a priority queue which is required in the A* algorithm

Terrain values

```
terrain[Openland] = 0.2
terrain[Roughmeadow] = 0.7
terrain[Easymovementforest] = 0.3
terrain[Slowrunforest] = 0.5
terrain[Walkforest] = 0.4
terrain[Impassiblevegetation] = -1
terrain[LakeSwampMarsh] = -1
terrain[Pavedroad] = 0.1
terrain[Footpath] = 0.1
terrain[Outofbounds] = -1
```

I have chosen negative values for Out of bounds, swamp/marsh/lake, and impassable vegetation as it is difficult to navigate. A rough meadow is navigable but more difficult due to trees and other obstacles so basically based on the obstacles, these values are set. Most convenient are paved roads and footpaths

The A* Algorithm will have a higher heuristic value for pixels having higher terrain modifier value thus making them less preferred over the pixels having low value.