Sara Savitz
U37713110
COT 4400 – Spring 2017

Project #3 Report

1.) For this project, I would use a weighted directional adjacency list to model the problem input. In the case of Tarzan and Jojo, we know Tarzan can only swing three or four squares in the direction indicated by the vertex. This means that each element in the adjacency list will have two neighbors at most. Given this information, the adjacency list is the optimal choice since it has a fast build time and the time complexity of its methods directly depend on the degree of each vertex. I would read in the input using a two dimensional array of size r*c; I would iterate through the array, and depending on the direction and number of squares away, add edges with the corresponding weight and direction.

2.) I would most likely use Dijkstra’s algorithm to find the shortest path. Using the adjacency list, Dijkstra’s algorithm would go to all neighbors of every vertex to create a minimum spanning tree from the starting point to the stopping point (JoJo’s location). Each edge visited would be stored in a predecessor list that Dijkstra’s algorithm would return once it reaches the stopping point. Since we are using a weighted adjacency list, Dijkstra’s algorithm works well in identifying the shortest path compared to Breadth First or Depth First Search.