Alan Rodriguez
u86831061
Project 3

1. I would model this problem using a directed, weighted, adjacency list to store all vertices and corresponding neighbors. To construct it I would first parse the input.txt as a 2d array with r rows, and c cols and then index through it. The vertices will be an iterative number based on it's position in the double for loop iterating rows and columns. This was there can be a unique number associated with numerous vertices. Based on the indexed direction of the vertex, and if the neighbors are valid, I would add neighbors to the adjacency list graph. The neighboring edges will have an associated weight of either 3 or 4 based on number of vines traveled. the edges will be identified by the int value of the vertices that are joined.

2. I will use Dijkstra's algorithm from the Boost library to find the shortest weighted path. All of Tarzan's possible moves will have already been stored in the adjacency list as edges in a stack. Since all edges have an associated weight of either 3 or 4, Dijkstra will pick the shortest path from Tarzan's adjacency list constructed. There will be cases where a vertex does not have 2 neighbors, and in such a case i will have to check if the neighbor to be is within bounds and allowed to be a part of the adjacency list. Jojo will have no out-edges, only in-edges. Dijkstras will create a minimum spanning tree from tarzan's start vertex to Jojo's vertex and find the minimum weighted path. However if Jojo is not part of Tarzan's adjacency list then Jojo will not be found.