1. **What type of graph would you use to model the problem input (detailed in the Section 3.1), and how would you construct this graph? (I.e., what do the vertices, edges, etc., correspond to?) Be specific here; we discussed a number of different types of graphs in class.**

I used an adjacency list to store the data from the input file. The adjacency list was implemented with an array of vectors. In other words, vectors were stored inside an array such as this: array[vector<int>, vector<int>, vector<int>,…]

For example, an adjacency list that looks like this

Shape

Description automatically generated with low confidence

would be stored like the following: array[<1,4>, <2,3,4>, <3>, <4>].

In this specific assignment problem, each letter (“N”, “E”, “S”, …) are the vertices and the edges are the connection / path between those. Also in this assignment, because Tarzan can travel only 3 or 4 spaces, each vertices had 0, 1, or 2 edges and neighbors.

Once I looped through every single letter / space, the whole graph was set up and the only step left to do was to find the path from Tarzan to Jojo.

1. **What algorithm will you use to solve the problem? Be sure to describe not just the general algorithm you will use, but how you will identify the sequence of moves Tarzan must take in order to reach the goal.**

I used the Breadth First Search algorithm to find the path. After the adjacency list was created with all the vertices and edges, running a simple BFS algorithm was enough to find the path to the destination. The adjacency list, however had to be designed carefully like explained above.

It might be helpful to think the adjacency list as a graph. You could also turn this graph into a BFS traversal tree for better visualization.