Problem 3: Implement Graph

In a collection of edges there are mininimal advantages. Searching for the nodes requires you to go through the entire collection for each one. It is quick to retrieve the number of edges and a simple way to understand the graph.It is however slow to retrieve the number of children a specific node has.

In an adjacency matrix it is easy to find the number of nodes and fast to retrieve a particular edge. It is difficult to represent an adjacency matrix to contain the different edge weights. It is more intuitive and better if the graph was not multiedged to use this type of implementation. It is slow to retrieve the number of children.

In an adjacency list, which I chose, it is easy to format the data as each node will be able to quickly retrieve its children. It is also simple to count the number of adjacencyLists there are and thus easy to retrieve the total number of nodes. The number of edges is a slow process, yet it is not slower than the collection implementation. If you use a dictionary to contain a node's adjacency list it reduces the search time and is the quickest implementation in terms of looking for children and overall.