Part 1

You have a fox, a chicken and a sack of grain. You must cross a river with only one of them at a time. If you leave the fox with the chicken he will eat it; if you leave the chicken with the grain he will eat it. How can you get all three across safely? Implement a solution to this problem.

How to do this? We can represent the problem using a graph $G = (V, E)$ and solve it using a search algorithm.

Let’s represent the state of the world with four bits $\langle b_0, b_1, b_2, b_3 \rangle$. Bit $b_0$ represents the location of the fox, $b_1$ represents the location of the chicken, $b_2$ represents the location of the grain, and $b_3$ represents your location. We will assume you are going from the left side to the right side of the river, thus for a particular bit $b_i, b_i == 0$ means the corresponding entity (fox, chicken, grain, or person) is on the left side of the river and $b_i == 1$ means the entity is on the right side. We can represent states by nodes in a graph $G = (V, E)$. Thus $V = \{\langle 0, 0, 0, 0 \rangle, \langle 0, 0, 0, 1 \rangle, ..., \langle 1, 1, 1, 0 \rangle, \langle 1, 1, 1, 1 \rangle\}$, or you could use the decimal representations of the bit vector and thus $V = \{0, 1, ..., 14, 15\}$. Legal transitions between legal states will be represented by undirected edges in that graph. A transition is legal if the person moves from one side of the river to the other and is accompanied by zero or one other entities. A state is not legal if the fox can eat the chicken or the chicken can eat the grain, as explained above, otherwise it is legal. Thus $(\langle 0, 0, 0, 0 \rangle, \langle 0, 1, 0, 1 \rangle) \in E$, corresponding to the person crossing the river with the chicken, while $(\langle 0, 0, 0, 0 \rangle, \langle 0, 0, 1, 1 \rangle) \notin E$, since $\langle 0, 0, 1, 1 \rangle$ is not a legal state (the fox eats the chicken).

We can then now find the solution to the problem by finding a path going from state $\langle 0, 0, 0, 0 \rangle$ to state $\langle 1, 1, 1, 1 \rangle$ in the graph. Your task is to implement this solution using a) breadth-first search and b) depth-first search.

Part 2

For Lab 6 you implemented a program to group words based on their similarities. Your task for this lab is to implement Prim’s algorithm to find the Minimum Spanning Tree of the graph that represents the words in Lab6Words.txt and their embeddings. Build an adjacency matrix representation of the graph, where vertex $i$ represents word $w_i$, and the cost of edge $(i, j)$ is given by $\text{sim}(w_i, w_j)$. Start your search with vertex 0 (corresponding to word able).

As usual, write a report describing your work.