Assignment 1

# Introduction:

The objective of this assignment was to solve the word ladder problem. The problem requires us to create ladders between any two words such that at each ‘rung’ of the ladder, we change one letter in the previous word until we finally reach the target word. Every word that makes up the ladder must be a part of the English lexicon.

# Tasks:

1. Create a library to solve the word ladder problem for any 2 words.
2. Create an executioner to find out the chains between all words in the above mentioned dictionary.
3. Generate a graph by connecting all words which do have a chain.
4. Calculate the minimum chains for all solvable words.
5. Find out the longest chain(s) formed after evaluating all solvable words.
6. Find out all the words that do not have a chain.
7. Calculate Frequency Distribution of the chain lengths.
8. Display your analysis and results on screen.
9. Include your analysis and results in the description file.
10. Use a Version Control System (VCS) to manage solutions.

# Algorithm:

The algorithm used is pretty straightforward. The word ladder problem can be solved easily using breadth first search. The algorithm I used is mentioned below for clarity:

Create an empty queue.

Add the starting word to the end of the queue.

while (the queue is not empty) {

Dequeue the first ladder from the queue.

if (the final word in this ladder is the destination word) {

Return this ladder as the solution.

}

for (each word in the lexicon of English words that differs by one letter) {

if (that word has not already been used in a ladder) {

Create a copy of the current ladder.

Add the new word to the end of the copy.

Add the new ladder to the end of the queue.

}

}

}

# Tasks completed:

* I was unable to calculate chains for all words due to lack of time/computing resources so I have created ladders for 1-5 letter words only.
* Created a frequency distribution graph for length of chains
* Uploaded the code to github
* The ladders created are included in separate file for 1-5 letter words in the format <num>ladders.csv . The lengths of the chains are included in a separate file called analysis.csv