Solution

This assignment is mainly to implemented a puzzle called a word ladder. This project can be divided into two parts. One is building a graph, and the other is using BFS (breadth first search) to find a path to convert word.

For the first part, I use a adjacent list to store the word list as a graph. The structure of the cell in the array is as same as the structure of the node in the list. I just store the index of respective word in the structure. To begin with, I will read the file and store all these words into a vector. The index of the vector is the identity of each word. While reading words, I will insert them into the array in the meantime. After that, I wrote a function to see which word is just one letter different with the current word, and store these words into the corresponding list. As a result, the graph built successfully.

For the BFS algorithm, I just implemented it according to textbook. I used a queue as a supplement. Also, I build two vectors: flag and previous. To record whether that node is visited and its parent node. First of all, I set all flags to false, and all previous to -1. Then, I initiate a queue using STL. After that, I get the index of the source word and the target word. I visit the source word. I set the flag of the visiting word to true and enqueue the index of the source word into the queue. Add pop it from the queue and delete it. After deleting it, I visit neighbors of that words. If its neighbors have not visited, then enqueue them and update the flag vector and the previous vector.

After that, I use the previous vector to tracing the process of the converting. I assume the user determine the k steps. If within the k steps, we get the target word, then it means it is successful. Otherwise, it will prompt “Cannot convert ”.