Shiv Chandra Kumar Program 4:alternative Word Ladder

Design decisions:

Graph: i am reading all the words and Building a graph using Adjacency list data structure because of scares graph here. I am reading all the node one by one from a vector where i stored the node after reading from file. After that for every node i am checking the word difference between the node and all other node. If there is only one word difference then i am taking the node as the adjacent node of the node we are checking now. Thus in my graph struct i have a array of pointer which points to all the adjacent nodes of the given vertex.

Queue: I do not see any use of priority queue here so i am using a normal queue for my operation here. there are normal queue operations defined here.

Algorithm and implementation decisions: I am using Breadth first search algorithm for finding minimum distance between any given two words. I am using following strategy:

1. I traverse through a graph start from the source node towards the destination node using breadth first search. on finding the destination my algorithm will exit the method and print the result.
2. i have taken into consideration that for a word length ‘k’ i will only check for ‘k’ steps for conversion while printing the result. when it exceeds i am printing the error message that i can not convert because of the length exceeding k.
3. ‘k’ steps is not including the source word. So this means a 6 letter word group will allow to print maximum 7 words including source and destination. If it take more than that then i will print can not convert error message as described above
4. also user has to put “exit” to terminate the program as it will keeps on asking for the input and print word ladder on providing one.

Output format is: GraphApplication Shiv$ ./wordLadder words

sample output is:

```
Enter Source(enter exit to terminate): 
jitter
Enter destination(enter exit to terminate): 
buster
Word Ladder is as follows: 
jitter
sitter
setter
senter
benter
bunter
buster
Enter Source(enter exit to terminate): 
custer
Enter destination(enter exit to terminate): 
jitter
Cannot Convert. It is taking more than 6 steps
```