CSCI/CMPE 3333 Assignment Four
Instructor: Zhixiang Chen

Team Work Policy: If you'd like, you work on this assignment in a team of at most two students. When submitting your assignment, each team only needs to submit one solution. But, all team members have to be listed.

Part A (100 points).

In this homework assignment, I would like you to use some combination of graph, hash, stack and queue data structures to solve the word ladder problem.

The Word Ladder Problem: Given a start word and an end word, find a shortest sequence of words (*shortest word ladder*) from the start to the end so that

- Each word is a legal word from a given dictionary
- The next word is obtained from the previous word by substituting exactly one letter.

If there is no such sequence, then say so. Note: Assume all words are in lowercase.

Example: Given a start word ***zero*** and an end word ***five***, here is the ladder:
zero => ***hero*** => ***here*** => ***hire*** => ***fire*** => ***five***
We assume that all the words are legal.

The Dictionary: The dictionary is given here:
http://faculty.utrgv.edu/zhixiang.chen/cs3333/3333/dic/hw2_dictionary.txt

The Shortest-Path Problem: Consider each word in the dictionary as a node. If one word is obtained from another word by substituting exactly one letter, then there is an edge connecting these two words. In this setting, the word ladder problem is equivalent to find a shortest path from one word (node) to another word (node). If there is no such a path, then say so.

If you can build a graph as mentioned above for the words in the dictionary, then you can directly apply a shortest-path finding algorithm to solve the word ladder problem.

The Challenge: When the dictionary is large, like the one you are given, it is not easy to build a graph. It may take too long or too much space so that it becomes unrealistic to build a graph.

How to Overcome the Challenge for unweighted letters: Use the graph at conceptually level. During the solution process, there is no need to build the complete graph. Use some sort of breadth-first or depth-first search strategy to find the word ladder. Note: For any node representing a word, without accessing to a fully-built graph for the dictionary, you can find its adjacent nodes (or words you can obtained from it through substituting exactly one letter). Think about how to do so, how to do so as fast as you can.

Caution: The above suggestion works only for unweighted words.

Caution: DO NOT PRINT OUT THE DATA FILES!

Specific Execution Time Requirement:
(1) The loading time of your program shall not be more than three minutes and the execution time for each pair of two words shall not be more than two minutes (consider the lab computers or compatible ones). Otherwise, no credit will be given to your program.

(2) Your program shall repeatedly ask the user to enter two words and then find the ladder for these two words or tell that no ladder exists. Each program will be tested by a fixed set of word pairs composed of

fishery ---- compute	*(length = 23)*
scroll ---- cloudy	*(length = 12)*
scroll ---- sundae	*(no ladder)*
computers ---- beautiful	*(no ladder)*
gimlets ---- theeing	*(length = 21)*

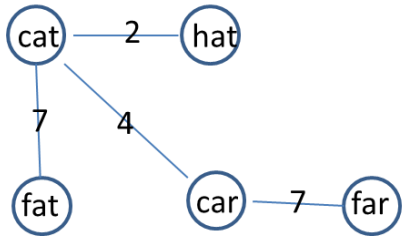
Part B (100 points).
Now we consider a weighted version of the Word Ladder Problem. A weight table is given below:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	2	1	2	4	2	4	9	1	6	2	1	4	9	2	5	1	2	3	9	0	1	1	5	9	3
b		0	6	3	9	4	7	1	1	8	9	7	5	9	5	6	3	2	3	5	7	7	8	6	8	9
c			0	7	2	7	7	2	9	3	3	1	7	1	8	7	5	6	8	5	2	8	8	7	7	5
d				0	5	7	8	1	5	4	7	6	1	9	0	1	1	4	9	5	5	5	7	5	6	2
e					0	6	2	0	1	7	4	4	1	3	8	5	9	4	9	7	4	4	3	8	0	5
f						0	8	1	8	3	8	2	3	9	6	2	9	3	9	6	2	4	6	3	8	8
g							0	2	7	1	3	2	9	7	3	4	2	7	9	5	3	2	4	1	0	0
h								0	1	2	3	0	5	7	1	4	4	7	3	6	4	7	1	0	0	2
i									0	6	1	3	5	0	2	1	5	6	5	2	8	2	9	5	7	6
j										0	4	1	6	6	7	1	5	8	7	7	5	7	2	7	4	6
k											0	2	2	7	7	2	5	2	0	5	9	4	1	9	3	5
l												0	9	1	9	4	1	1	9	0	1	6	9	3	9	4
m													0	0	2	9	9	5	2	2	1	3	6	9	7	5
n														0	1	2	3	5	7	9	1	5	3	7	3	6
o															0	2	7	9	2	4	6	1	3	5	5	5
p																0	0	0	5	1	6	2	2	4	1	1
q																	0	8	2	6	1	1	7	8	9	9
r																		0	7	4	9	3	2	7	0	6
s																			0	2	4	4	2	2	5	4
t																				0	5	1	6	1	5	6
u																					0	1	6	2	2	0
v																						0	4	7	5	7
w																							0	4	2	8

x																						0	4	6
y																							0	2
z																								0

Note that this table is symmetric with respect to diagonal line, so only entries in the upper right portion are given.

For example, the graph below shows edge weights as determined by the table



The Problem: Your need to re-do Part A with the weight word graphs as determined by the weight table. For a pair of the start word and the end word, this time you have to find the shorted weighted ladder from the start to the end. Specific execution time requirements from Part A apply to Part B, and the same test word pairs will be used.

Due Date: The due date will be given via Blackboard.

Warning: Any submission one week after the due date will not be accepted.

How to submit your work?
Please upload your source program files and your test results to Blackboard.