ECM5605 (5076) S’18: Homework 01

DUE: 2018/02/21 23:59:59

[Problem 1] (100%) Wavy Word Ladder Puzzle

“Word Ladder Puzzle” is a word game also named as “Doublet” or “Word-links”. The goal of this puzzle is to get people to appreciate the work and artistry that go into creating good word ladders. “Word Ladder” is invented on the Christmas day in 1877 by Lewis Carroll. It is such a beautiful game with a long history that even quoted in a novel, “Pale Fire”. “Some of my records are: hate—love in three, lass—male in four, and live—dead in five (with ‘lend’ in the middle),” says the narrator in “Pale Fire”.

The rule of “Word Ladder” is quite simple. The player is given **a start word** and **an end word**. In order to win the game, the player must change the start word into the end word progressively, according to the following three rules:

1. Each step consists of a single letter substitution, OR one letter is added/subtracted from the word.
2. Each transformed word must exist in the dictionary.
3. The transformation sequence**(s)** should be the **shortest**.

Example 1.

Start word: COLD
End word: WARM
Dictionary: {**CORD**, **CARD**, **WARD**, **WARM**, CORN, WEED}

Ladder: COLD → CO**R**D → C**A**RD → **W**ARD → WAR**M**

Example 2.

Start word: CARER
End word: NONE
Dictionary: {**CARER**, **CORER**, **CORNER**, WARD, **CONER**, **CONE**, **NONE**}

Ladder: CARE → CARE**R** → C**O**RER → COR**N**ER → CO~~**R**~~NER → CONE~~**R**~~ → **N**ONE

In this homework, you are going to find the shortest word ladder(s) according to the rules. The main code structure is provided, and you only have to finish the function: **findLadders** in hw1.cpp.

Input: begin word, end word, and a dictionary.
Output: Shortest Ladder(s).

Input file <.in>

Format:

```
<beginWord> <endWord>
<Dictionary Words>
```

Example:

```
$ cat test1.in
CAN NOT
ERIC HAF PEAG WAUR THIR IW UP ABRI TA JUN OSLO ALA KOR SIDA
HULU FED SYE IRE MISS SIMS EAU LOWE ASH MACK KORY REFL NERD
UTU ISN MIKI IAMB ADMI SEAT BLEA ABO MUD NAPE FOW RYAL REEN
BOH RCD RAY SCAD HIM EA SODA RF SF ALN YIRM AZ AIL BRR SEPG
```

Output file <.out>

Example:

```
$ cat test1.out
=====    Answer Report    =====
CPU Run time: 0
Mem Usage:        1948
Word ladder number: 1
Word ladder length: 4
ans[0]: CAN -> CAT -> COT -> NOT
```

Note that the main structure is provided in hw1.cpp, so you **DON'T** have to parse the input file yourself. Moreover, you **DON'T** have to measure CPU and memory yourself, because we've already done that for you in main().

Things you need to provide:

1. “hw1.cpp” with **findLadders** implemented.
2. A simple report describing how you implement the function.

How we evaluate your code:

1. Accuracy: Check if you find the shortest ladder(s)
   a. Number of shortest word ladders you found.
   b. The length of the word ladders.
2. Performance: (Execution time)
   You are **NOT encouraged** but welcome to use parallel programming. Note that we will accumulate CPU clocks in all the threads/tasks you use.
3. Memory Usage
   We will measure the resident set size, which is your RAM usage including your code size.

RSS = Heap size + MetaSpace + OffHeap size.

Good luck and have a nice holiday ~