Word Ladders

Thore Husfeldt

April 9, 2015

Description

Find paths between given words among the five-letter words of
English.

How words connect

We define a directed graph of words as follows. There is one node for
every 5-letter word. There is an arc from v to w if each of the last four
letters of v appears in w. For example, there is an arc from *yodel* to
lodes, but not from *lodes* to *yodel* because the latter contains no S. On
the other hand, there is an arc from *sharp* to *graph* and back. All four
letters have to appear with repetitions, so there is an arc from *where*
to *ether* (both Es appear) but not to *retch* (E appears only once). As an
example, here's a pretty long path in the graph:

climb → blimp → limps → pismo → moist → stoic → ioctl → colts → lotsa → stoae → oaten → neath → hated → dated
→ dater → rater → tread → dared → dread → drear → rarer → reran → arena → earns → snarf → franc → narco →
orcas → scare → raced → decaf → fecal → eclat → talcs → clasp → psalm → slams → small → llama → lamas → amass
→ smash → shame → hames

Figure 1: words-10.

Requirements

You build a directed graph representing the adjacency structure of
the words in the list called `words-5757.txt`, and run BFS on it. The
input to your algorithm is two words, like `climb` and `hames`. Your
algorithm returns the words on a shortest path from the first word to
the second.

   The data directory contains a number of test inputs with known
distances, your algorithm must work correctly on those.[1]

   If you can, make the running time of your algorithm $O(n + m)$,
where n and m refer to the number of nodes and arcs in the un-
derlying directed graph. In particular, you're not supposed to use
quadratic time in n to build the graph, even though it's tempting.[2]

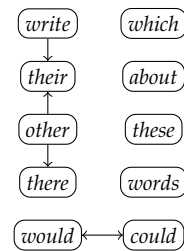[1] There can be many shortest paths, so it
makes no sense to specify the words on
the path.

[2] Of course, for dense graphs, $O(n + m)$
would be quadratic time anyway. But
the input graph is not dense.

Deliverables

1. The source code for your implementation

2. A report in PDF. Use the report skeleton in the `doc directory`.

Figure 2: words-250