Word Ladders Report

Adrian Brink, Asger Pedersen, Simon Flachs, Troels Møller

April 14, 2016

Results

Our implementation produces the expected results on all input–output file pairs.

On input `words-5757.txt`, a shortest path from aargh to zombi of length 10 is the following:

> aargh graph parch chard hoard radon nomad dogma amigo gizmo zombi

Implementation details

We build the graph's edges by iterating over all input words, generating all alphabetically ordered 4-letter-permutations from the word. Each permutation is added to a hashmap as a key with the word as a element in a linked list. If other words produce the same permutations, they are added to the corresponding linked list of the given permutation.

The running time for the graph construction is:

$$O(n \cdot w_1 \cdot w_2 \cdot (w-1) \cdot (w-1) \cdot \log(w-1)$$

n * w is to build the hashmap, the rest is to create and sort the permutations alphabeticaly.

The running time for searching will of course be dependent on what words are used as start an ends points, and what words are in the graph. In general the complexity can be expressed as the branching factor for each node (number of adjacent nodes) to the power of the depth, d, of a path from a start word to the end word. The average branching factor is equal to the number of edges m, divided by the number of nodes, n.

$$b = \left(\frac{m}{n}\right)^d$$