Word Ladders Report

Daniel Odenbrand

April 6, 2016

Results

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

On input `words-250-test.in`, a shortest path from equal to value cannot be found, but a path from shows to ready of length 7 is found:

```
shows->whose->horse->store->other->heart->trade->ready
```

Implementation details

I build the graph's edges by iterating over all nodes twice and comparing the words by iterating through it's letters, thus taking $O(n^2w)$.

The total running time of our implementation (including graph construction and traversal) is $O(n^2w)$, but the BFS algorithm takes only $O(n+m)$.

It is possible to build the graph by first sorting all letters of every word and creating each possible combination of strings of length 4 (there are only 5 since the string is sorted and will be compared with sorted strings only). We then use a map where the strings of length 4 act as keys, mapping to the word. If two words would have the same key, they are both stored in a vector in the map. Then we iterate over all words once again to take the last 4 letters of the word, sort them, and try the mapping. If the key exists then a directed edge should be from the current word to the one(s) stored in the map.

This construction of the graph will take $O(nw)$ for creating the map, and another $O(n+m)$ to generate all edges for a total of $O(n(w+1)+m)$.