Word Ladder Report

Thomas Keralla Høpfner-Dahl, Domenico Villani & Zander Havgaard

Our implementation produces the expected results on all input– output file pairs, with no exceptions, see appendix.

On input words-5757.txt, a shortest path from aargh to zombi is of length 10 with the following path:
aargh to zombi (10): aargh -> graph -> parch -> chard -> hoard -> radon -> nomad -> dogma -> amigo -> gizmo -> zombi

Implementation details

We build the graph's edges by iterating over all inputs from the file and comparing each input string against all other inputs with our check function. This gives us a quadratic running time for the graph construction.
The Breadth First Search algorithm implemented concatenates an index in front of each string input and thus runs in O(v+e).
The total running time of our implementation (including graph construction and traversal) is thus the creation of the graph and running the BFS in O-notation: O((n^n) + (v+e)).

In order to improve the running time for the very large inputs like 104500, we have made a parallelized version of Digraph which is included under DigraphThread.java. This means that the time for the input 104500 is lower due to optimization, however the amortized time for BFS is the same.

Appendix

```
➜  sol git:(master) ✗ java WordLadder words-10.txt words-10-in.txt
Time for creating graph = 0.007
Graph done, beginning BFS
other to there (1):  other -> there
other to their (1):  other -> their
could to would (1):  could -> would
would to could (1):  would -> could
there to other (-):  not connected
about to there (-):  not connected
Clean time: 0.002
Total BFS time: 0.009
Time total with findSourceIndex = 0.012
➜  sol git:(master) ✗
```

```
➜  sol git:(master) ✗ java WordLadder words-50.txt words-50-in.txt
Time for creating graph = 0.011
Graph done, beginning BFS
other to there (1):  other -> there
there to where (1):  there -> where
large to earth (2):  large -> great -> earth
there to there (0):  there
every to earth (-):  not connected
other to where (2):  other -> three -> where
where to other (-):  not connected
Clean time: 0.0019999999999999983
Total BFS time: 0.006999999999999999
Time total with findSourceIndex = 0.012
```

```
➜  sol git:(master) ✗ java WordLadder words-250.txt words-250-in.txt
Time for creating graph = 0.061
Graph done, beginning BFS
equal to value (-):  not connected
value to equal (1):  value -> equal
alike to field (-):  not connected
write to river (3):  write -> tried -> drive -> river
shows to ready (7):  shows -> whose -> horse -> store -> other -> heart -> trade -> ready
Clean time: 0.0040000000000000036
Total BFS time: 0.008999999999999994
Time total with findSourceIndex = 0.012999999999999998
➜  sol git:(master) ✗ java WordLadder words-50.txt words-50-in.txt
```

```
➜  sol git:(master) ✗ java WordLadder words-500.txt words-500-in.txt
Time for creating graph = 0.099
Graph done, beginning BFS
there to where (1):  there -> where
these to where (3):  these -> sheet -> three -> where
place to wheat (8):  place -> clear -> layer -> years -> raise -> ideas -> shade -> death -> wheat
woods to woods (0):  woods
woods to grown (-):  not connected
whole to wheat (13):  whole -> holes -> solve -> vowel -> lower -> wrote -> other -> heart -> tears -> raise -> ideas -> shade -> death -> wheat
homes to wheat (13):  homes -> moves -> solve -> vowel -> lower -> wrote -> other -> heart -> tears -> raise -> ideas -> shade -> death -> wheat
Clean time: 0.006000000000000005
Total BFS time: 0.01200000000000001
Time total with findSourceIndex = 0.015
```

```
➜  sol git:(master) ✗ java WordLadder words-5757.txt words-5757-in.txt
Time for creating graph = 3.435
Graph done, beginning BFS
oared to bared (1):  oared -> bared
oared to oared (0):  oared
daily to hones (6):  daily -> inlay -> lawny -> yawns -> weans -> ashen -> hones
meter to micro (5):  meter -> ester -> tiers -> emirs -> scrim -> micro
prawn to bread (4):  prawn -> warns -> nares -> sabre -> bread
zombi to aargh (-):  not connected
aargh to zombi (10):  aargh -> graph -> parch -> chard -> hoard -> radon -> nomad -> dogma -> amigo -> gizmo -> zombi
alpha to gamma (-):  not connected
bring to beers (6):  bring -> grins -> siren -> inter -> nerts -> terse -> beers
altos to duets (4):  altos -> louts -> ousts -> suets -> duets
abaca to papas (-):  not connected
Clean time: 0.0770000000000004
Total BFS time: 0.0860000000000003
Time total with findSourceIndex = 0.10199999999999987
```

```
➜  sol git:(master) ✗ java WordLadder words-104500.txt words-104500-in.txt
inputlength: 104500
Threadpool: 50
THREADED!!! INPUT_SIZE:104500 adj_size: 104500
Done assigning index: 0.138
makeConnection time: 499.315
Time for creating graph = 499.725
Graph done, beginning BFS
sykur to socka (4):  sykur -> rusky -> ukosy -> yocks -> socka
avdel to otawa (5):  avdel -> lavde -> davte -> etova -> votaa -> otawa
inget to whizz (-):  not connected
fancy to dalek (4):  fancy -> caney -> yaken -> nekad -> dalek
puten to vitae (3):  puten -> nietu -> utvei -> vitae
inget to eslau (4):  inget -> tegna -> genua -> nuesa -> eslau
bodil to vyryj (8):  bodil -> idylo -> dobyl -> obryl -> lybry -> bryzy -> zrywy -> wyryj -> vyryj
Clean time: 6.247000000000014
Total BFS time: 6.252000000000066
Time total with findSourceIndex = 6.305999999999983
```