

EE422C Project 3 (Word Ladder) Team Plan

Peter Yu pmy89
Jia-luen Yang jy8435
Fall 2016

Both of us met up on Monday/Tuesday/Wednesday to work on the project. All coding was done with both members present. We spent approximately 7 hours together in the same room. Test cases were discussed and compiled through Google docs and also in person. Most of the work was done on Jia-luen Yang's laptop since GitHub messed up our code several times when we merged the edits. Everything worked as we expected.

However, there were many mistakes and confusions about conflicts in the PDF guide and what the TAs said on Piazza. In fact, the `printLadder()` function would sometimes not work in certain conditions. We made sure to ask on Piazza several times and discuss with other students in the same class.

Final testing was done together. We made sure to check for stack overflow and the 30 second time limit for DFS. The time limit was never a problem. We encountered stack overflow due to our initial pseudo code for DFS. We declared too many unnecessary variables within the recursion. As a result, the stack overflowed. We fixed it by rewriting DFS completely with minimal local variables.

For BFS, the only key part of getting the shortest link between the two words is actually reading from the queue. We came up with many solutions but only one made the most sense and worked as we expected. The algorithm is to take the bottom of the queue and find the related words from the top. Once we found that related word, it then became the bottom of the queue and we keep doing this until we reach the first word of the queue. That word would then be the start word. By using this method, we are sure to get the shortest link between two words. One of our test cases also made sure to test BFS for reversing the order of the words. We still get the same number of links. That's how we know that our BFS is getting the shortest links.