Project 3 Word Solver in Java

Kyle Ryan, University of Rochester

November 2015

1 Summary

In this project, we implemented a word solver program. The program takes a text file with a table of n by n characters with words hidden in the table. Given a dictionary of words, the program loops through every possible direction and cell in the table to determine whether the iteration of characters is a word. The program uses a hash table that uses a standard hash function to take each word in the dictionary and store it. The run time of a hash table is as follows: Search: O(1), Delete: O(1), Insert: O(1), Space: O(n). A hash table is useful for storing information with a known size.

2 Algorithm Summary

Algorithm Summary:

1. Iterate through each cell in the table.

2. At each cell, iterate in 8 directions (up, up right, right, down right, down, left down, left, up left).

3. At each direction find the max length possible of words before the end of the table.

4. Loop from the length down to 0.

5. At each time, check the hash table to set if the word is in the dictionary.

3 Run-time Analysis

To check the run time of this algorithm, I included a simple illustration,

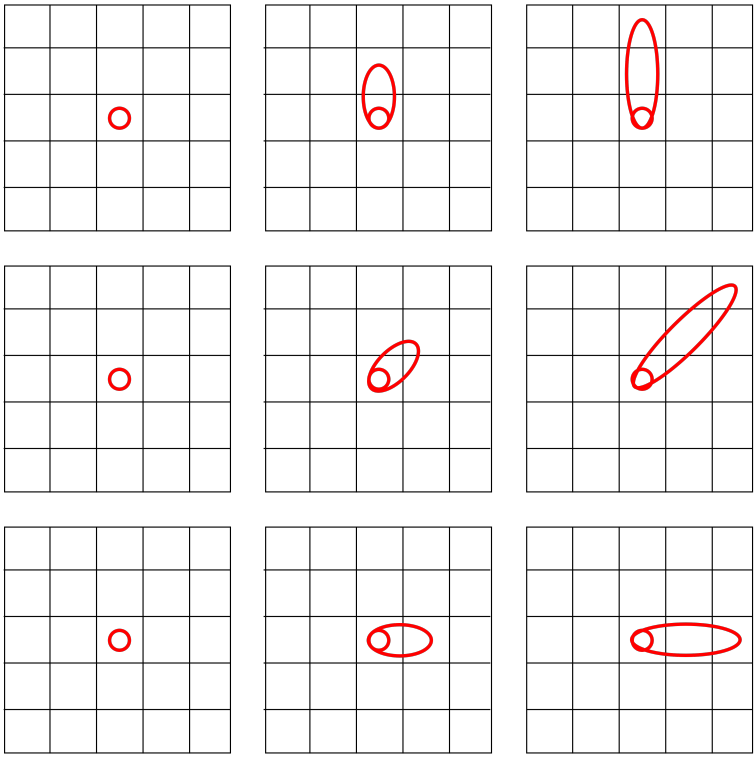

Figure 1: Short illustration about algorithm used to find all words.

By checking in 8 directions, we know that the run-time analysis will include $O(8n)$. Since the table size is N by N then the final calculated run time of this algorithm will be $O(n^3)$.

4 How to Run

To run this program, I included all of the classes in a single file. The main method is in the p̈roject3c̈lass. To run, you can use terminal and type the following:

$ javac project3.java

$ java project3 dictionary.txt input.txt output.txt

The format of the input for the .class file is java project3 [dictionary] [input] [output]. The program will automatically export all of the found words to an output text file. It will also print out all of the found words into the console.

5 Extra Credit

5.1 Error handling

For extra credit, I implemented error handling. This checks to make sure that the inputted text file is in fact (NxN). If it is not N by N, then the program will quit and exit.

5.2 GUI

For some more extra credit, I implemented a GUI to show all thr found words. I made the WordGrid class to do this. However, I did not have enough time to program a method to show the found words. It was difficult to figure out how each JLabel could be represented to show found words.