

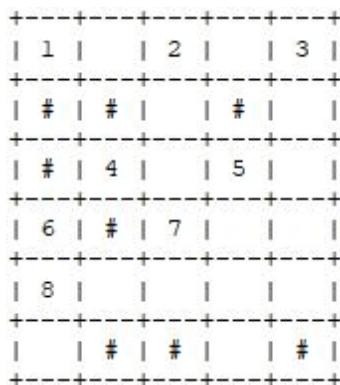
Ben Bissantz
bbiss001@odu.edu
11/12/2020

MSIM 580 Assignment Three: Crossword Puzzle Designer

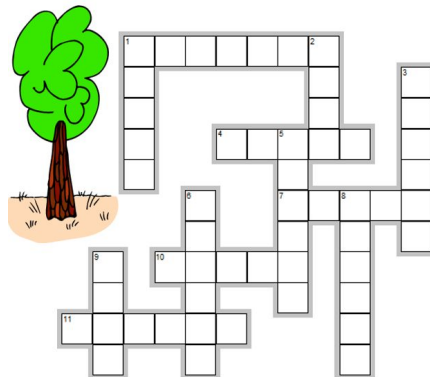
Overview

The program to implement a backtracking algorithm to design a crossword puzzle consists of two classes. The main class, Crossword.java includes the main method, the backtracking algorithm methods, a dictionary sorting method, and is responsible for file input and output. The other class, Word.java includes information about a single word within the crossword puzzle including location, direction, other words that cross the word, and a list of possible words from the dictionary.

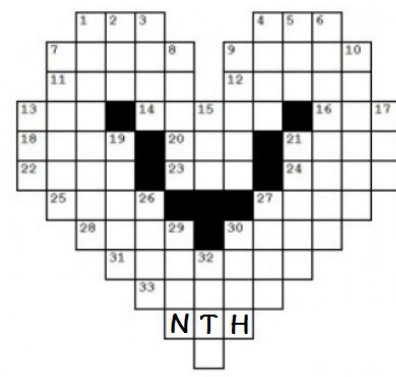
This program was implemented to solve each of the three puzzles included in the assignment, referred to in the program as the Simple Puzzle, Tree Puzzle, and Heart Puzzle. Using a simple backtracking algorithm only, a backtracking algorithm with forward checking, a backtracking algorithm with forward checking and minimum remaining values, and a backtracking algorithm with forward checking and degree heuristics.



Simple Puzzle



Tree Puzzle



Heart Puzzle

Program Execution

When running the program, the user is first prompted to select a puzzle file. Puzzle files are formatted with the character zero representing spaces in the puzzle for words to be filled in and the character space representing areas outside of the puzzle that will not be filled in. In place of zeroes lowercase letters may be included, for example to add initials to the Heart Puzzle.

Once the user has selected a puzzle file the user is prompted to select a dictionary file. The user may select the original Words.txt for this assignment, a pre-sorted version of Words.txt, the list of 15 words associated with the Simple Puzzle or a pre-sorted version of the

list of 15 words. Additionally the user is prompted to add custom words to the program which may be required if initials have been added to the Heart Puzzle or if the input file contains other prefilled sections.

000000	0000000	000 000	ere ohm
0 0	0 0 0	00000 00000	atoll spoor
00000	0 0 0	00000 00000	these table
0 000	0 00000 0	000 00000 000	ate easel eve
00000	0 0 0	0000 000 0000	fire sea shew
0 000	0 00000	0000 000 0000	tree eel tire
00000	0 0 0 0	0000 0000	earn rile
0 0	0 00000 0	0000 0000	lied yell
	0 0 0 0	0000000	evacuee
	000000 0	00000	aural
	0 0 0	ben	ben
		0	w

Simple Puzzle Input

Tree Puzzle Input

Heart Puzzle Input and Output

Lastly the user is prompted to select search heuristics. The user may choose to use a simple backtracking algorithm without additional heuristics, a backtracking algorithm with forward checking, a backtracking algorithm with forward checking and minimum remaining values, or a backtracking algorithm with forward checking and degree heuristics.

```

C:\Users\User\Desktop\MSIM 580 AI\Assignment 3\MSIM 580 A3\crosswordDesign>set path=C:\Program Files\Java\jdk-15.0.1\bin
C:\Users\User\Desktop\MSIM 580 AI\Assignment 3\MSIM 580 A3\crosswordDesign>javac Crossword.java Word.java
C:\Users\User\Desktop\MSIM 580 AI\Assignment 3\MSIM 580 A3\crosswordDesign>cd C:\Users\User\Desktop\MSIM 580 AI\Assignment 3\MSIM 580 A3
C:\Users\User\Desktop\MSIM 580 AI\Assignment 3\MSIM 580 A3>java crosswordDesign.Crossword
Please select a file containing a puzzle. Enter digits as follows for the corresponding file.
1 - Heart Puzzle
2 - Tree Puzzle
3 - Simple Puzzle
4 - Cat Puzzle
5 - Select a Custom Puzzle
1
Please select a dictionary file containing a list of words. Enter digits as follows for the corresponding file.
1 - Standard Dictionary Word List
2 - Pre-Sorted Standard Dictionary Word List
3 - Simple Dictionary Word List Containing 15 Words
4 - Pre-Sorted Simple Dictionary Word List Containing 15 Words
5 - Select a Custom Dictionary Word List
1
Adding a custom word may be required if the puzzle file contains words that are not in the dictionary word list.
For example if initials are included in the Heart Puzzle the initials will need to be added here if they are not also a word.
Would you like to add a custom word.
1 - Yes
2 - No
1
Please enter the word. Format supports lower case letters a-z only.
Do not include special characters, upper case letters, or numbers.
ben
ben added.
Would you like to add a custom word.
1 - Yes
2 - No
2
Please select the search heuristics to use.
1 - Simple Backtracking Algorithm with No Heuristics
2 - Forward Checking Only
3 - Forward Checking with Minimum Remaining Values
4 - Forward Checking with Degree Heuristics
3
Reading Inputs ... Creating Puzzle ... 13 by 12 Puzzle Created ... Finding Words ... 42 Words Found ... Finding Cross Points ... Cross Points
Found ... Creating Dictionary ... 14551 Words Added to Dictionary ... Inputs Complete.

```

User Input at the Start of the Program

Program output is saved to a text file named after the input file and search heuristic used. The output file contains the completed puzzle, program statistics, and a word list. The program statistics include the total number of comparisons the backtracking algorithm made while adding words to the puzzle; the time spent searching for a puzzle solution and the time preparing data prior to starting to search; and a summary of the dictionary used to find the solution. The word list includes the location, length, and word for all of the words in the crossword puzzle.

Backtracking Algorithm and Heuristics

The first portion of this program was the implementation of the backtracking algorithm. The backtracking algorithm successfully solved both the Simple Puzzle and the Tree Puzzle in a very short amount of time but it was unable to solve the Heart Puzzle without the addition of heuristics.

A forward checking heuristic was developed to verify that adding a word did not result in any other words having no possible words. This prevented the program from continuing to search along invalid paths. The forward checking heuristic was developed using a linked list of all possible words assigned to each word in the crossword puzzle. As words are added to the puzzle, the lists of any crossing words are updated by removing any invalid words. If this results in a word having no possible words in its list, the forward checking heuristic removes the word that was added and resets the possible word lists of all affected words.

In addition to forward checking the possible words list is also used to get the word with the minimum remaining value (MRV) which is selected as the next word to update when MRV is used as a heuristic. After a word is added to the puzzle and successfully fulfills the forward checking heuristic the possible words lists for the remaining words are compared to find the word with the MRV.

Alternatively to MRV, the next word to update may optionally be selected using degree heuristic. The degree heuristic finds the word that crosses with the most other words that have not been added to the crossword puzzle. This is found by subtracting the number of characters in a word that have already been assigned from the number of words that cross with a word.

Other Puzzles

To provide an additional test to the program a custom Cat Puzzle was developed. A brief search of puzzles online revealed that many common crossword puzzles use proper nouns, abbreviations, initials, or multiple words. All of these word choices are not supported by the dictionary file or the program in its current form; therefore, a custom puzzle was created. Although the puzzle has less words than the heart puzzle it took much longer to solve, possibly due to the use of two letter words which have much fewer possible words than longer words.

The Cat Puzzle was solved using forward checking with minimum remaining values after 35,106,825 comparisons in 2 hours, 5 minutes, 31.990 seconds.

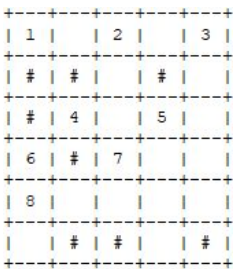
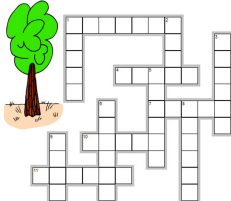
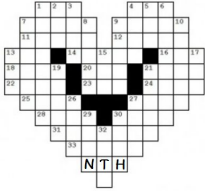
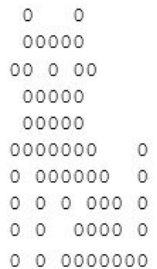
```

0 0          e a
00000       spoor
00 0 00     at c me
00000       attar
00000       trade
0000000 0   peevish  l
0 000000 0   e footed e
0 0 0 000 0   t o s let e
0 0 0000 0   e i alee r
0 0 0000000  r l anomaly
Cat Puzzle Input and Output

```

Results

Although the Simple Puzzle and the Tree Puzzle were both solved in a matter of milliseconds regardless of heuristic choice, the more difficult Heart Puzzle and Cat Puzzle were only solved when using forward checking combined with minimum remaining values. The results of solving the four puzzles are summarized in the table below.

Puzzle	 Simple Puzzle	 Tree Puzzle	 Heart Puzzle	 Cat Puzzle
Backtracking Only	199 Comparisons 7 ms	42 Comparisons 48 ms	Did not complete within 3 hours	Did not complete within 3 hours
Forward Checking	24 Comparisons 5 ms	42 Comparisons 55 ms	Did not complete within 3 hours	Did not complete within 3 hours
Forward Checking MRV	24 Comparisons 6 ms	105 Comparisons 67 ms	5,275,229 Comparisons 17 m 20.678 s	35,106,825 Comparisons 2 h 5 m 31.990 s
Forward Checking Degree Heuristics	24 Comparisons 5 ms	42 Comparisons 56 ms	Did not complete within 3 hours	Did not complete within 3 hours

The Simple Puzzle had a significantly smaller search space with a dictionary of only 15 words and only 8 words total. The Tree Puzzle has no more than three words crossing any word and has no adjacent words in the same direction. For both the Simple Puzzle and Tree Puzzle the limited differences in results make it difficult to compare heuristic selections. The Simple Puzzle required the most comparisons using the simple backtracking algorithm only possibly because this puzzle has a significant number of points where words cross which would benefit from the use of heuristics. Alternatively the Tree Puzzle required the most comparisons when using forward checking and minimum remaining values. This could be due to the limited number of cross points which could cause word length to have a greater effect on the minimum remaining value calculation resulting in a less efficient result.

The Heart Puzzle and Cat Puzzle both have substantially more points where the words cross which creates more dependencies which make the puzzles more difficult to solve and require additional heuristics. For these puzzles forward checking combined with minimum remaining values was the best heuristic selection.