Project 1 Readme drewdiguglielmo

Version 1 9/11/24

A single copy of this template should be filled out and submitted with each project submission, regardless of the number of students on the team. It should have the name readme_"teamname"

Also change the title of this template to "Project x Readme Team xxx"

<table>
<tr><td>1</td><td>Team Name: drewdiguglielmo</td></tr>
<tr><td>2</td><td>Team members names and netids: Drew DiGuglielmo, ddigugli</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects: Implementing a polynomial time 2-SAT solver</td></tr>
<tr><td>4</td><td>Overall success of the project:</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: I would estimate roughly an hour and a half of preparation and outline/understanding of the project, 7 hours on code, and 2 hours on the graph code/writeups</td></tr>
<tr><td>6</td><td>Link to github repository: https://github.com/ddigugli/theoryproject1</td></tr>
<tr><td>7</td><td>List of included files (if you have many files of a certain type, such as test files of different sizes, list just the folder): (Add more rows as necessary).
<table>
<tr><td>File/folder Name</td><td>File Contents and Use</td></tr>
<tr><td colspan="2">Code Files</td></tr>
<tr><td>2SAT_code_drewdiguglielmo</td><td>2SAT code implementing a polynomial time 2SAT solver</td></tr>
<tr><td>graphplot_code_drewdiguglielmo</td><td>Code to show the plot from my outputfile</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>2SAT_check_drewdiguglielmo</td><td>Test file given in canvas to use to test code</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td>outputfile_csv_drewdiguglielmo</td><td>Output file from using the test file given in canvas</td></tr>
<tr><td colspan="2">Plots (as needed)</td></tr>
<tr><td>graphoutput_drewdiguglielmo</td><td>Screenshot of the graph from running my graph test code</td></tr>
</table></td></tr>
<tr><td>8</td><td>Programming languages used, and associated libraries: The programming language</td></tr>
</table>

	used was python and I used sys, csv, and time to access command line arguments, writing to a csv file, and calculate the execution time in microseconds for my main 2SAT code. For the graph, I also used python, and used the pandas and matplotlib.pyplot for data analysis and plotting.
9	Key data structures (for each sub-project): The first key data structures for the 2SAT code were lists (line.split(), wff.append(line), poss), which were used to handle lines of data, literal clauses, and variable assignments. I also used dictionaries (values, dictstack), to store the variable truth assignments and to track branching points and the state of variable assignments in the case of backtracking. Finally, I used a csv writer object file which was to write to the csv file to store the results.
10	General operation of code (for each subproject): The main operation code starts with a fileread function which reads the input file and does line-by-line processing and calls my wff function each time a line begins with the letter ‘c’. The satisfiability function determines whether a wff is satisfiable using backtracking and propagation of variable assignments. It begins by assigning values to variables and checking constraints, and backtracking if a dead end is reached. The wff function parses each wff, splitting into literals and clauses. The propagate function updates the assignments based on the current assignments and will modify the truth values based on constraints. A final line is used to summarize the wff count, number of sat vs unsat, which are written to the csv file.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code: I used the provided sat test case in canvas. It helped me verify the correctness of my backtracking and overall logic of checking sat vs. unsat. While coding, I used test cases in the file provided, but a shortened version so that the runtime would be faster for me to check the result and determine where my code logic should be tweaked.
12	How you managed the code development: I split the code into many functions in order to remain organized and be able to tackle one aspect at a time. There are many helper functions called within to aid with the organization, which also helped for debugging purposes as the individual components were separated.
13	Detailed discussion of results: Overall, my code does what it was designed to do successfully, mainly through the satisfiability function which measures the execution time, which is then written to the CSV file. My graph has a good trend that aligns with what you should see for this type of problem. By analyzing the csv file, my graph, and the overall development of the code, I can verify the correctness of my code.
14	How team was organized: Since individual, team organization was not considered.
15	What you might do differently if you did the project again: Potentially work with a partner, even though it is more work, it would aid in times when I got stuck and did not know what to do while staring at the code.
16	Any additional material: N/A