Project TeamworkTemplate

Version 1 9/11/24

A ***separate copy*** of this template should be filled out and submitted by each student, regardless of the number of students on the team. Also change the title of this template to “Project x Teamwork <team> - <netid>”

<table>
<tr><td>1</td><td>Team Name: Rademacher</td></tr>
<tr><td>2</td><td>Individual name: Grace Rademacher</td></tr>
<tr><td>3</td><td>Individual netid: grrademac</td></tr>
<tr><td>4</td><td>Other team members names and netids</td></tr>
<tr><td>5</td><td>Link to github repository: https://github.com/grademac/Theory_Project1</td></tr>
<tr><td>6</td><td>Overall project attempted, with sub-projects: Bin Packing Problems #1 (Coin Problem)</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>coinstester_Rademacher.py</td><td>dumbSat equivalent for the bin packing problem in which it solves for the combination of coins for a given amount. It reads in a file of targets and coin amounts and outputs in both terminal and an output file if it’s satisfiable and the amounts for each coin. It also outputs the execution time. The solver skips a case if it takes longer than 5 minutes to solve.</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>knapsackTestCaseGen_Rademacher.py</td><td>Creates a file of randomly generated targets and coin values. The user can adjust how many types of coin values are generated and how many cases in total are generated. These values are outputted into a file.</td></tr>
<tr><td>Test_Rademacher (folder)</td><td>Test files generated from knapsackTestCaseGen.py for each of the number of coins cases</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td>Timeresults_Rademacher (folder)</td><td>Time results from each of the number of coins cases (2-5 representing how many types of coins were available) –</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>used to create graphs</td></tr>
<tr><td>SATresults_Rademacher (folder)</td><td>Satisfactory results from each of the number of coins cases (2-5 representing how many types of coins were available) – used to create graphs</td></tr>
<tr><td>results_Rademacher (folder)</td><td>Output results from each of the number of coins cases (2-5 representing how many types of coins were available)</td></tr>
<tr><td colspan="2">Plots (as needed)</td></tr>
<tr><td>Project 1 Graphs_Rademacher</td><td>Shows number of types of coins vs microseconds, number of types of coins vs log microseconds, and the number of types of coins vs the average number of microseconds</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>Individual Student time (in hours) to complete: ~5 hours</td></tr>
<tr><td>9</td><td>Your specific activities and responsibilities: everything</td></tr>
<tr><td>10</td><td>What was personally learned (topic, programming, algorithms): This project definitely illustrated just how quickly a program slows down once you get to 3 coins and above. This just showcases how difficult it is to solve 3Sat and above problems. This was also one of the first projects I’ve done that didn’t have a direct path for me to take so I learned more about my personal programming style and how to integrate test cases into my code.</td></tr>
<tr><td>11</td><td>How the team was organized, and what might be improved: Since it was just me, the team wasn’t organized in any particular way but I do think I could improve my timeline in working through a project. I kept switching on which project I wanted to do, so next time I just need to be more decisive.</td></tr>
<tr><td>12</td><td>Any additional material:</td></tr>
</table>