Project 01 Readme Team slama

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: slama</td></tr>
<tr><td>2</td><td>Team members names and netids: Sophie Lama, slama</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects: Hamiltonian Path solver</td></tr>
<tr><td>4</td><td>Overall success of the project: I would consider the project a success, it accomplished what it needed to. The program implements a brute-force algorithm to determine whether a graph contains a Hamiltonian path or not. It works for graphs of varying sizes, taking much longer for the bigger graphs.</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: 6-8 hours</td></tr>
<tr><td>6</td><td>Link to GitHub repository: https://github.com/siscalie/theory_project01</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). 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>hamiltonian_path_ existence_slama.py</td><td>Program that opens a TXT file describing various graphs (nodes with edges), determines if each graph contains a Hamiltonian path or not, then outputs the results in a CSV file.</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>hamiltonian_path_ test_cases_slama.txt</td><td>Text file with pairs of lines representing graphs (one line provides a list of nodes in a graph, the second line provides a list of edges in that graph)</td></tr>
<tr><td colspan="2">Output Files</td></tr>
</table>
</td></tr>
</table>

	<table><tr><td>output_slama.csv</td><td>CSV file that contains information on whether it is True or False that a graph in the input file contains a Hamiltonian path, the size of the graph (the amount of nodes in the graph), and the time it took to determine if there was a Hamiltonian path (in ms).</td></tr><tr><td colspan="2">Plots (as needed)</td></tr><tr><td>plots_image_slama.png plots_excel_slama.xlsx</td><td>plots_image_slama.png is an image of the execution time plot. plots_excel_slama.xlsx is the Excel file used to create this plot.</td></tr></table>
8	Programming languages used, and associated libraries: **Python, with itertools (to create permutations of tuples), time (to get timestamps when calculating execution times), ast (to quickly convert strings/lines from a file into lists), and collections (when creating a default_dict for the graph).**
9	Key data structures (for each sub-project): **I represented each graph as a dictionary, with keys being integers (nodes) and values beings lists (lists of edges connected to the key node).**
10	General operation of code (for each subproject): **The code begins by opening the file with graph information. In a while loop that reads lines from the file, the graph data is extracted from the lines, the start time is recorded, we attempt to build a Hamiltonian path in the graph (this is a brute-force operation involving a for loop that attempts to build a Hamiltonian path for every possible permutation of the nodes), then the end time is recorded. Finally, we write the results of the program to a 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 TXT file provided in the Project 01 Files folder on the Canvas page. This file was easy to open and read in Python, and it was useful since it said whether each graph was Hamiltonian or not.**
12	How you managed the code development: **The first part of the code I wrote was opening and reading files with graph information. I then tried to convert that information into formats that would be easy to use with a brute-force algorithm. The final and most difficult part was developing the brute-force algorithm.**
13	Detailed discussion of results: **I would say I'm happy with the results of the project. The brute-force method takes a long time with the bigger graphs, but it seems accurate as far as I can tell.**
14	How team was organized: **I was a solo team on this project.**

15	What you might do differently if you did the project again: **I would try to test a lot more graphs than the test graphs provided in the TXT file. I ended up adding more graphs to the TXT file to get a clearer plot of the exponential execution times, but if I had more time I'd add even more graphs.**
16	Any additional material: