

Project 2 Readme Team Andrew

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”

1	Team Name: Team Andrew																
2	Team members names and netids: Andrew Cotaj - acotaj																
3	Overall project attempted, with sub-projects: NTM Trace																
4	Overall success of the project: Very Successful																
5	Approximately total time (in hours) to complete: ~9-10 hours																
6	Link to github repository: https://github.com/AdrewC2026/Project2-TOC																
7	<div>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><th>File/folder Name</th><th>File Contents and Use</th></tr><tr><td colspan="2">Code Files</td></tr><tr><td>./src/ntm_tracer.py</td><td>File providing functionality for NTM Tracer utilized by main.py *All other files (enrtpoint.py, main.py, and helper files were not changed)</td></tr><tr><td colspan="2">Test Files</td></tr><tr><td>./src/input/a_plus.csv ./src/input/composite.csv ./src/input/equal_01s.csv ./src/input/ktape.csv ./src/input/ntm_n1n.csv</td><td>Provided file - accepts string of as Provided file - accepts composite numbers Provided file - accepts string of equal number of 0s & 1s Included in Forked Repo - Different Topic Included in Forked Repo - Different Topic</td></tr><tr><td colspan="2">Output Files</td></tr><tr><td>./aplus_aaa.</td><td>Photo showing output of running program with aplus.csv and input aaa</td></tr><tr><td>./composite_111111.</td><td>Photo showing output of running program with</td></tr></table></div>	File/folder Name	File Contents and Use	Code Files		./src/ntm_tracer.py	File providing functionality for NTM Tracer utilized by main.py *All other files (enrtpoint.py, main.py, and helper files were not changed)	Test Files		./src/input/a_plus.csv ./src/input/composite.csv ./src/input/equal_01s.csv ./src/input/ktape.csv ./src/input/ntm_n1n.csv	Provided file - accepts string of as Provided file - accepts composite numbers Provided file - accepts string of equal number of 0s & 1s Included in Forked Repo - Different Topic Included in Forked Repo - Different Topic	Output Files		./aplus_aaa.	Photo showing output of running program with aplus.csv and input aaa	./composite_111111.	Photo showing output of running program with
File/folder Name	File Contents and Use																
Code Files																	
./src/ntm_tracer.py	File providing functionality for NTM Tracer utilized by main.py *All other files (enrtpoint.py, main.py, and helper files were not changed)																
Test Files																	
./src/input/a_plus.csv ./src/input/composite.csv ./src/input/equal_01s.csv ./src/input/ktape.csv ./src/input/ntm_n1n.csv	Provided file - accepts string of as Provided file - accepts composite numbers Provided file - accepts string of equal number of 0s & 1s Included in Forked Repo - Different Topic Included in Forked Repo - Different Topic																
Output Files																	
./aplus_aaa.	Photo showing output of running program with aplus.csv and input aaa																
./composite_111111.	Photo showing output of running program with																

	built the NTM_Tracer class, starting with a basic BFS loop to verify tree expansion.. Testing was conducted sequentially, ensuring the simple a_plus machine functioned perfectly before integrating and debugging the more memory-intensive logic required for the composite and equal_01s machines.
13	Detailed discussion of results: The results highlighted the distinct performance characteristics of NTMs, where the a_plus machine showed a high degree of nondeterminism (~1.8) due to constant branching, while equal_01s remained mostly deterministic (~1.1) with targeted guess points. The traces confirmed that the BFS approach was superior to DFS for this assignment, as it successfully found shallow solutions (like in composite.csv) without getting trapped in infinite recursion depth. Furthermore, the successful execution of the equal_01s traces demonstrated that the tape management logic correctly handled the "infinite blank tape" assumption by dynamically padding the string representation whenever the head moved beyond the existing input boundaries.
14	How team was organized: Individual Project - N/A
15	What you might do differently if you did the project again: If I were to redo this project, for cleanliness sake, I would optimize the memory management by replacing the "list of lists" history structure with a standard queue for BFS and a separate dictionary for parent pointers, as the current method retains every computed configuration and consumes exponential memory on deep trees. I would also implement a visited set to hash and prune redundant configurations (identical state and tape content) to prevent processing duplicate branches, which would significantly increase the depth the simulator could reach before hitting memory limits.
16	Any additional material: N/A