Program Structures and Algorithms
Spring 2024

NAME: Gautham Venkata Krishna Prasad
NUID: 002249901
GITHUB LINK: https://github.com/gauthamkris7neu/INFO6205Assignment

Task:

Assignment 4 (WQUPC) : Implement height-weighted Quick Union with Path Compression.

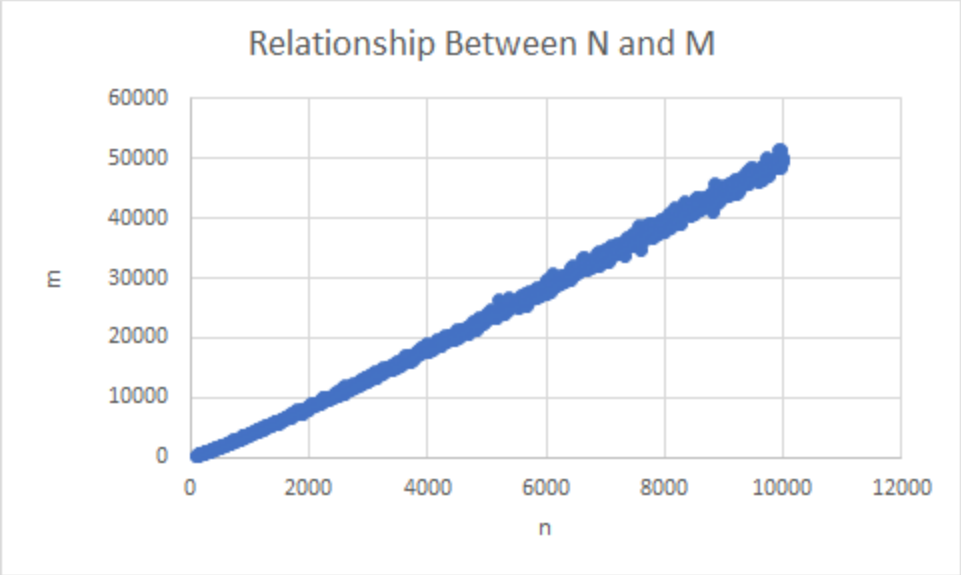
Relationship Conclusion:

In conclusion, the relationship between the number of objects nn and the number of pairs mm generated to fully connect all objects in a union-find structure, optimized with path compression and height-weighted unions, can be approximated by a simplified empirical relationship of m ≈ nlogn. This relationship strikes a balance between the inherent efficiency of the algorithm and the practical considerations of random pair generation and the process of connecting disjoint sets. The logarithmic factor in this relationship accounts for the diminishing efficiency gains as the size of the network increases, reflecting the union-find algorithm's capability to maintain near-constant time operations for union and find commands through path compression and height balancing. This simplified relationship provides a practical framework for estimating the computational effort required to achieve a fully connected network, offering valuable insights for applications involving dynamic connectivity, clustering, and network analysis, where understanding the scalability and performance of underlying algorithms is crucial.

Evidence to support that conclusion:

The data for this was take using the Union Find Client implemented as a part of the assignment:

https://northeastern-my.sharepoint.com/:x:/g/personal/venkatakrishnapras_g_northeastern_edu/EdH0jxn_JWtNhDauc1qNri8BteX8cPaU37pWq-Owb3GY_w?e=U58KF6



The graph depicts a clear trend where the number of pairs ‘m’ needed to connect all objects grows as the number of objects ‘n’ increases. This relationship is sub-linear, as the curve's slope increases with ‘n’, indicating that ‘m’ grows at a rate that is proportional to ‘n’ multiplied by a logarithmic factor of ‘n’. The trend is consistent with the expected performance of an optimized union-find algorithm, where path compression and height-weighting contribute to an efficient scaling of operations.

Unit Test Screenshots: