

Jayden Younger

Ming Li

CS 2123 Data Structures

24 September 2024

CS 2123 Programming Project 1

DAG Name	Number of vertices n	Induction-based time (Ti)	DFS-based time (Td)	Speedup (Ti/Td)
DAG1	5000	29.531 ms	5.948 ms	4.965 ms
DAG2	10000	61.315 ms	6.630 ms	9.248 ms
DAG3	50000	196.115 ms	17.065 ms	11.492 ms
DAG4	100000	2988.001 ms	34.793 ms	85.881 ms
DAG5	200000	18079.736 ms	53.036 ms	340.893 ms

DFS-Based Topological Sort vs. Induction-Based Topological Sort

Both algorithms have a time complexity of $O(V + E)$, but the DFS-based Topological Sort outperforms and is faster than the Induction-Based Topological Sort compared to our results. A few factors may cause this major performance difference:

- The number of vertices and edges may affect the performance since we are using a minimum of 5000 which is still a dense number of vertices. If we use a significantly less amount of vertices, the performance difference might not be less noticeable.

- Dfs-based topological Sort has more memory overhead from the number of recursive calls and stack management while Induction-Based Topological Sort is more iterative.

Impact of Problem Size on Performance

$$\textit{Speedup} = \textit{Induction based time} / \textit{DFS based time}$$

Using this formula and implementing it in our code, we see that the speed-up time increases as we increase the number of vertices. In large graphs, the induction-based sort may show more significant speed advantages, but that advantage may change if the number of vertices decreases. This can come down to the graph structure, recursion depth, and memory overhead.