Report: Analysis of Topological Sort

A graph is formally defined as the ordered pair G = (V, E) where V is a set of nodes and E is a set of edges. It would thus be a good strategy to look at what effects changing V and E have on our algorithms for Topologically Sorting a graph using Depth First Search and Kahn's Algorithm for Topological Sort. Here on, we will use |V| to signify the number of vertices and |E| to signify the number of edges.

***Note:** All the values plotted on the graphs are averages taken over five inputs for the same combination of | V | and | E|. We use clock ticks in order to measure the running time of our algorithms on various inputs.*

|V| is constant, |E| is variable:

I kept |V| constant at 2000 and used daggen.c to vary the probability of getting an edge between two nodes from 0 to 100.

As we can see from the graph that topological sorting using DFS and Kahn's method are both linear with respect to the number of edges, O(|E|). Although the slope of the plot of Kahn sort is much greater than the slope of DFS algorithm, thus signifying better performance of DFS as compared to Kahn Sort.

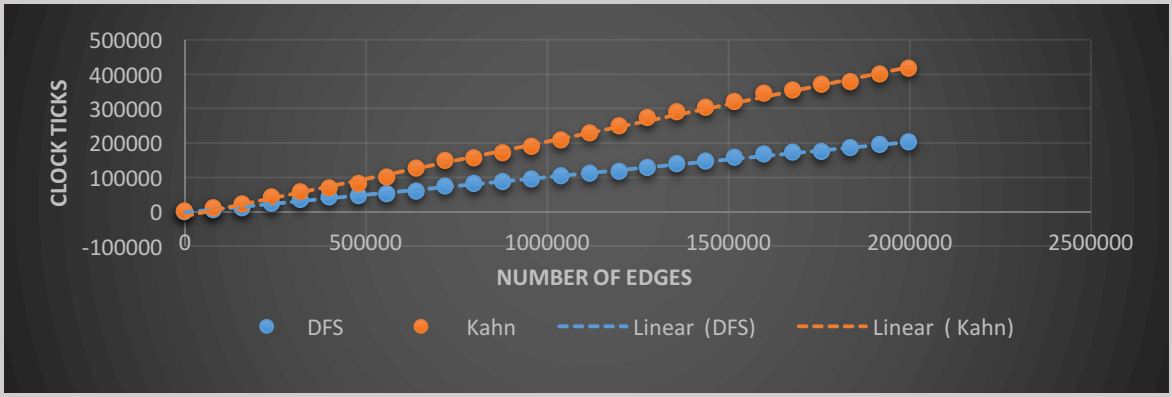


Figure 1: |V| constant, |E| variable

|E| is constant, |V| is varying:

In order to vary |V|, I fixed |E| at a value of 10000 and varied |V| from 1000 to 27000 in steps of 1000. Our findings indicate that Topological Sorting, irrespective of the algorithm is linear with respect to the number of vertices i.e. O(|V|). In this case too, DFS performs better than Kahn sort.

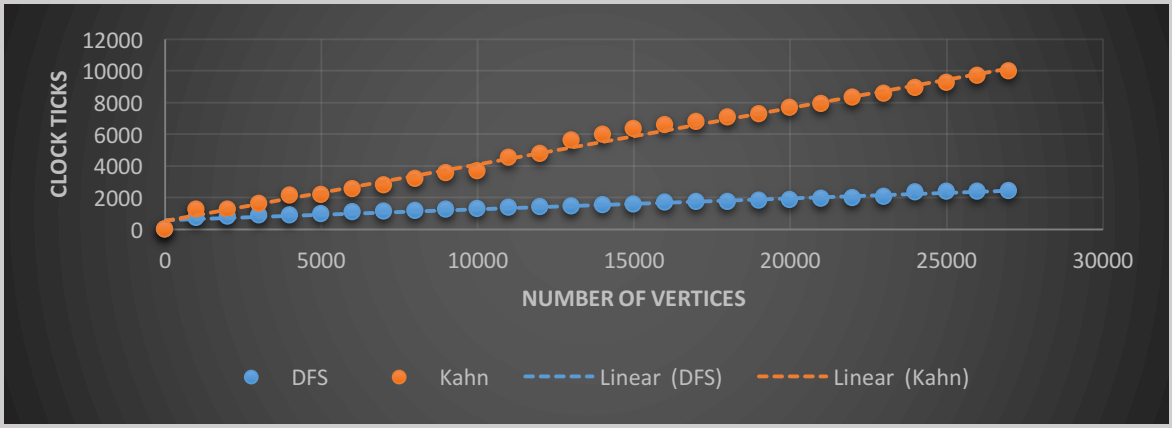


Figure 2: |E| constant, |V| variable

|V| and |E| are both changing:

Since it has already been confirmed that topological sorting (irrespective of the method) is linear with respect to |V| and |E| independently. We randomize |V| and |E| in order to change them simultaneously. The graph again shows linear behavior. Thus, we can conclude that Topological Sorting using both DFS and Kahn's method is linear showing average case complexity, O(|V| + |E|).

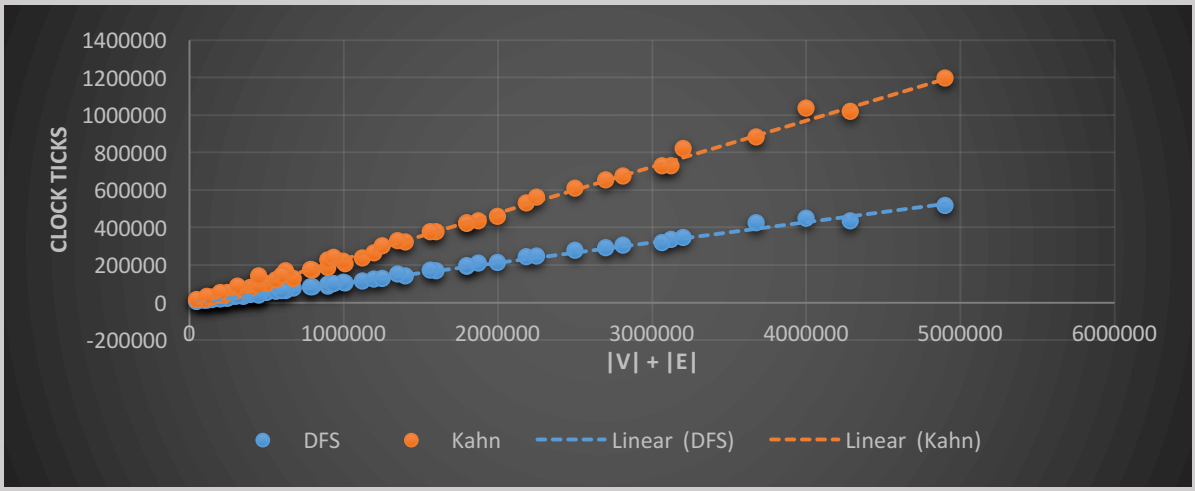


Figure 3: |E| and |V| variable

Making sense of the findings:

Topological sorting using both Kahn and DFS should theoretically be O(|V| + |E|) (from Wikipedia). The three sets of experimental inputs that we use are also pointing to a complexity of O(|V| + |E|).

Looking into the DFS algorithm, we mark each vertex as being unvisited in O(|V|) time and then visit each vertex using the dfs_visit() function(we only visit a vertex if it has not been visited before). Thus, ensuring we do not visit a vertex more than once or go down an edge twice, giving us a complexity of O(|V| + |E|) for DFS. There are seemingly no inputs that would cause DFS to lose its linearity or act undesirably.

The analysis for Kahn Sort is a little bit more complicated as it be complexity, O(|V| + |E|) but may not turn out to be so if we do not implement it properly. We first find the length of the list of incoming edges to each edge and store it in an array. This operation costs O(|E|). During the process of Kahn's algorithm, we can just decrement the number of edges coming into the vertex in order to simulate an edge deletion, the decrement costs us O(1) but if we actually deleted the edge we would have an added complexity of O(|V|) for each edge in the graph leading to loss of linearity. We also don't add new vertices to the tail of the list of topologically sorted elements and instead add to the head since it is more economical (O(1) instead of O(|V|)) and then reverse later at a cost of O(|V|). Thus using a couple of tricks we can ensure retention of Kahn sort's linearity.

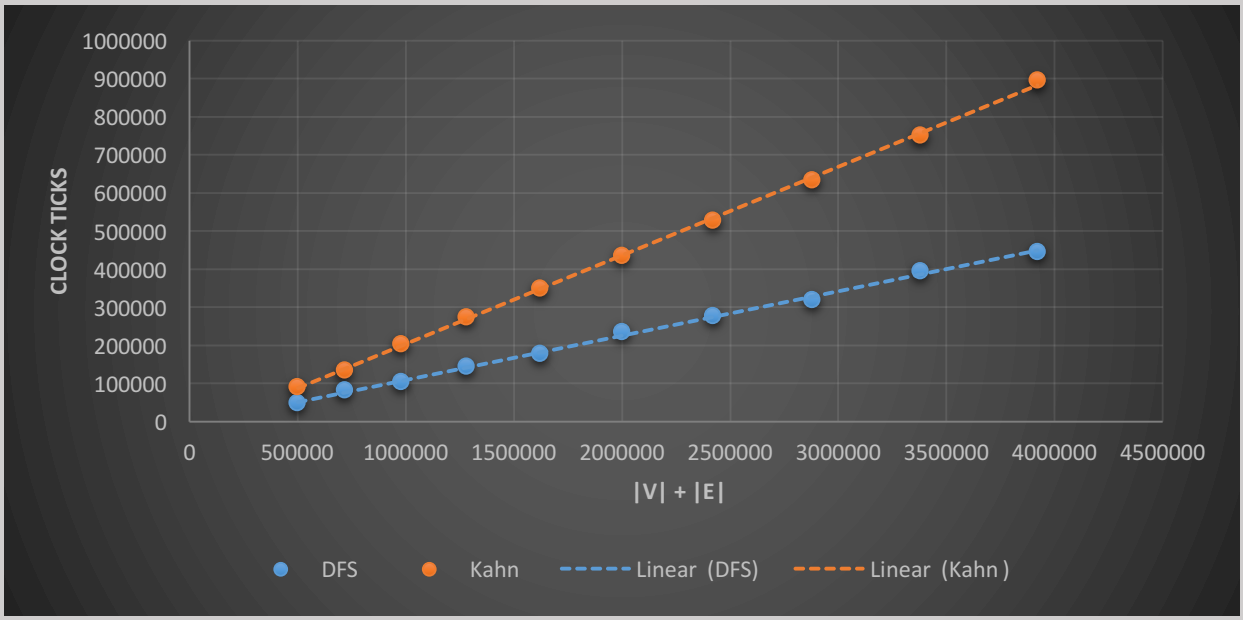
Kahn VS DFS:

A comparison of two algorithms is warranted if they are designed to execute the same task. DFS sort has a few advantages over Kahn sort, for example in case we have a cyclic graph as input DFS would find it as soon as we encounter a back edge but Kahn sort would execute the whole program create a topologically sorted list only to find the invalidity of the input thereafter. Moreover, the gradient of DFS is smaller than Kahn sort and the is much faster than Kahn Sort for larger inputs.

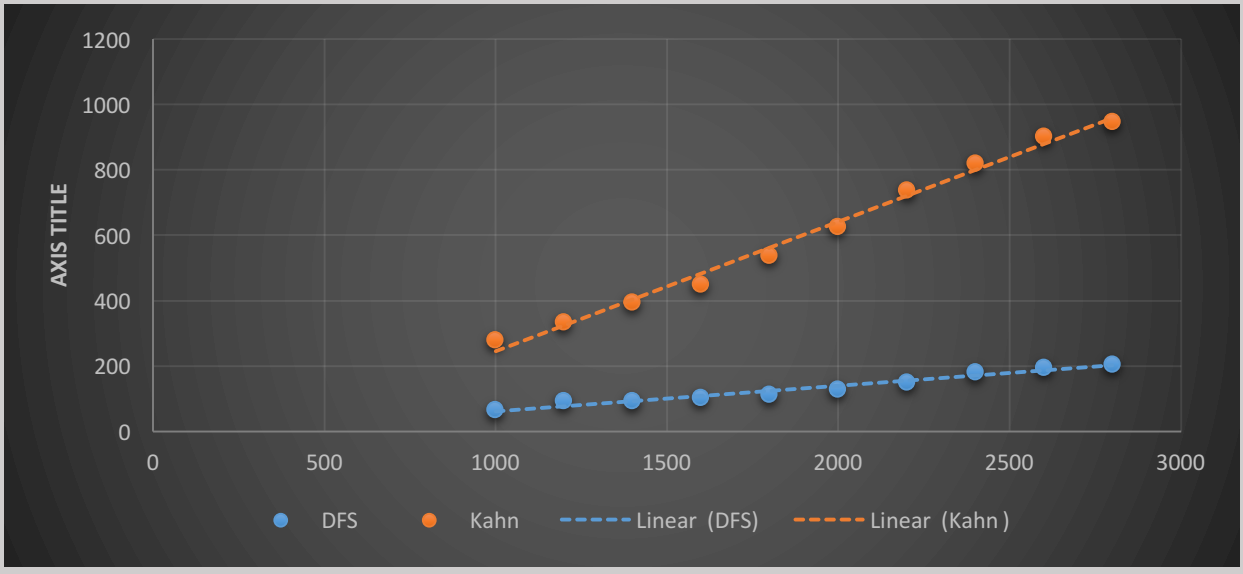
APPENDIX:

A:

Graph 1: 100% edge probability given to daggen.c for different vertex sizes:



Graph 2: 0% edge probability given to daggen.c for different vertex sizes:



Both the extreme cases above show linear behavior enforcing my claim that both DFS and Kahn's sort are linear with respect to |V| and |E|.

B: References:

- Wikipedia(https://en.wikipedia.org/wiki/Topological_sorting)