# Topological sort

*Kevin Woo*

Someone who don’t know how to count

## Introduction

**Topological sort** is usually done to a directed acyclic graph (dag)/ A topological sort of a dag$G = ( V, E)$ is a linear ordering of all its vertices that if$G$ contains an edge $(u, v)$, then $u$ appears before $v$ in the ordering. (I did this in Google in order to show a dependency graph for asynchronous workflows).

## Algorithm

Topological sort is basically a complexity of$O( V + E)$ given that we can do it within 1 pass of a DFS. As we "finish" a vertex (find $u.\ f$), we append to the front of a linked-list. It is basically the vertices sorted in descending order by $u.\ f$ for every vertex $u$.

## Exercises

**22.4-1 Show the ordering of vertices produced by TOPOLOGICAL-SORT when it is run on the dag of Figure 22.8, under the assumption of Exercise 22.3-2.**

| vertex | discovered | finished |
|---|---|---|
| m | 1 | 20 |
| n | 21 | 26 |
| o | 22 | 25 |
| p | 27 | 28 |
| q | 2 | 5 |
| r | 6 | 19 |
| s | 23 | 24 |
| t | 3 | 4 |
| u | 7 | 8 |
| v | 10 | 17 |
| w | 11 | 14 |
| x | 15 | 16 |
| y | 9 | 18 |
| z | 12 | 13 |