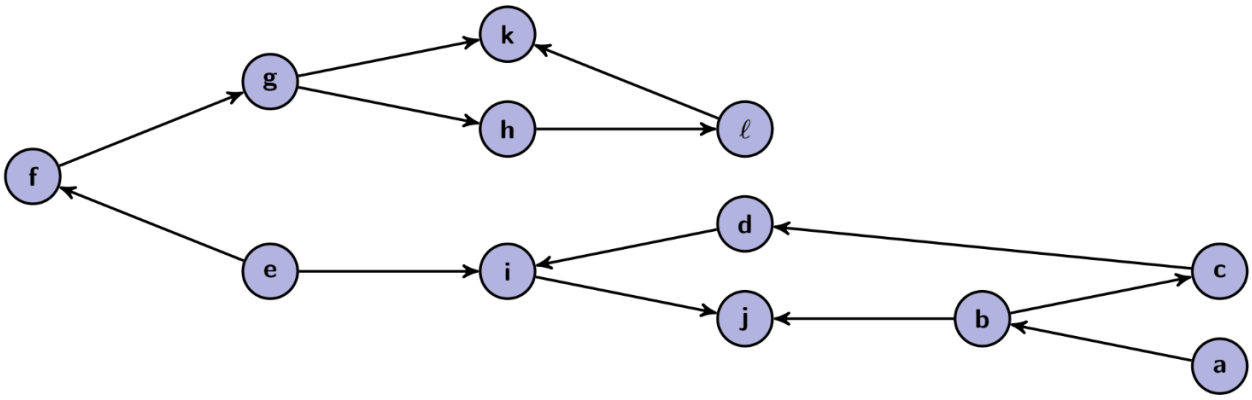
CSE 332: Data Structures and Parallelism

Section 8: Graphs

0. It Rhymes with Flopological Sort

Consider the following graph:



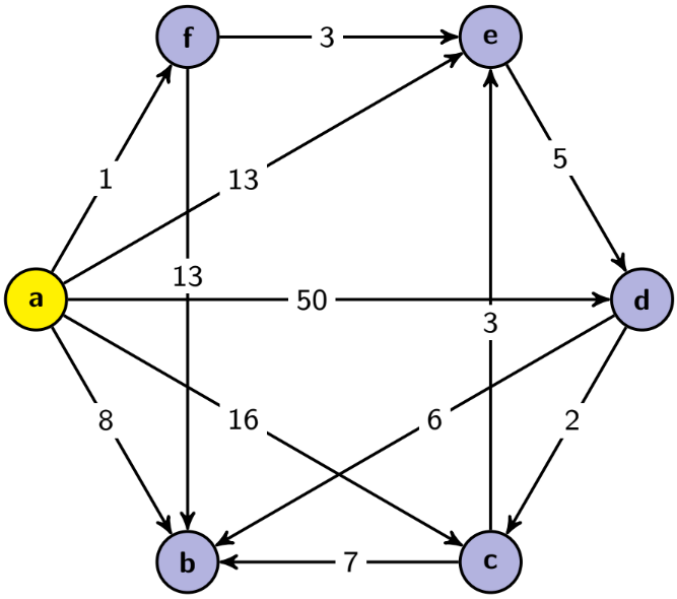
a) Does this graph have a topological sort? Explain why or why not. If you answered that it does not, remove the **MINIMUM** number of edges from the graph necessary for there to be a topological sort and carefully mark the edge(s) you are removing. Otherwise, just move on to the next part.

For the remaining parts, work with this (potentially) new version of the graph.

b) Find a topological sort of the graph. Do not bother showing intermediary work.

1. Velociraptors

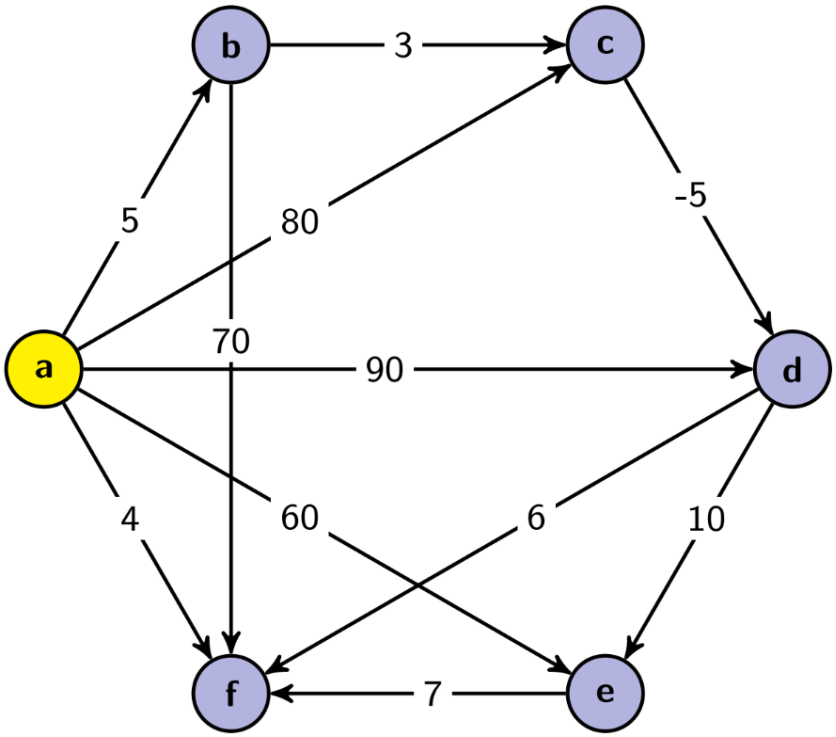
Consider the following graph:



Suppose that you are at **a** and you are planning your escape from a bunch of hungry velociraptors (edge weights represent the expected number of velociraptors you will meet on this path). Run Dijkstra's Algorithm to **find the lengths of the shortest paths** (fewest number of velociraptors met) from **a** to each of the other vertices. Remember to store the path variable and **list the order vertices are added to the known set**.

2. Better Find the Shortest Path Before It Catches You!

Consider the following graph:



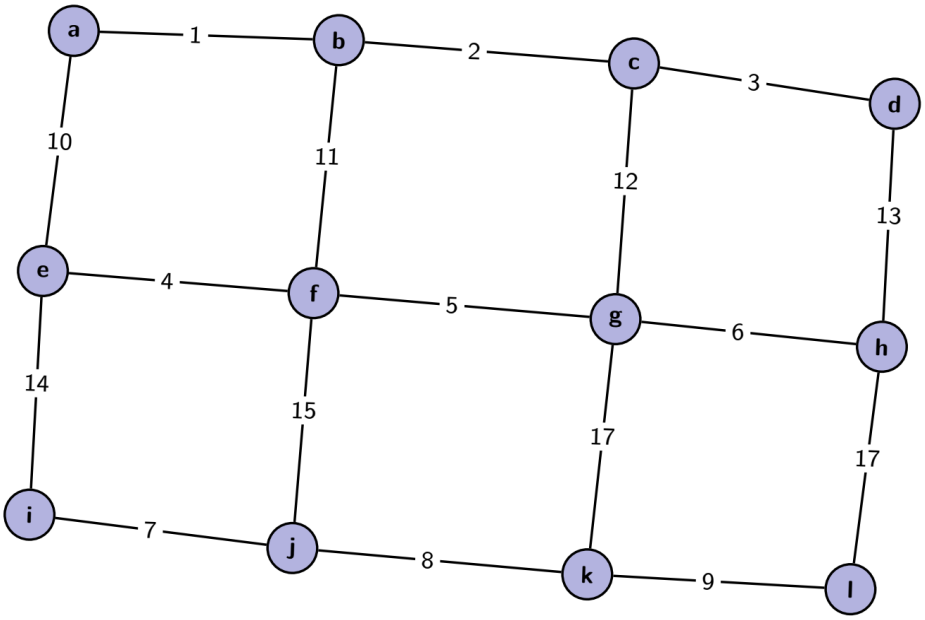
a) Use Dijkstra's Algorithm to find the lengths of the shortest paths from a to each of the other vertices. Show your work at every step.

b) Are any of the lengths you computed using Dijkstra's Algorithm in part (a) incorrect? Why or why not?

c) Explain how you would use Dijkstra's Algorithm to recover the actual paths (rather than just the lengths).

3. LMNST!

Consider the following graph:



a) Find an MST of this graph using both of the two algorithms we've discussed in lecture. Make sure you say which algorithm you're using and show your work.

b) Using just the graph, how can you determine if it's possible that there are multiple MSTs of the graph? Does this graph have multiple MSTs?

c) What is the asymptotic runtime of the algorithms that you used to compute the MSTs?