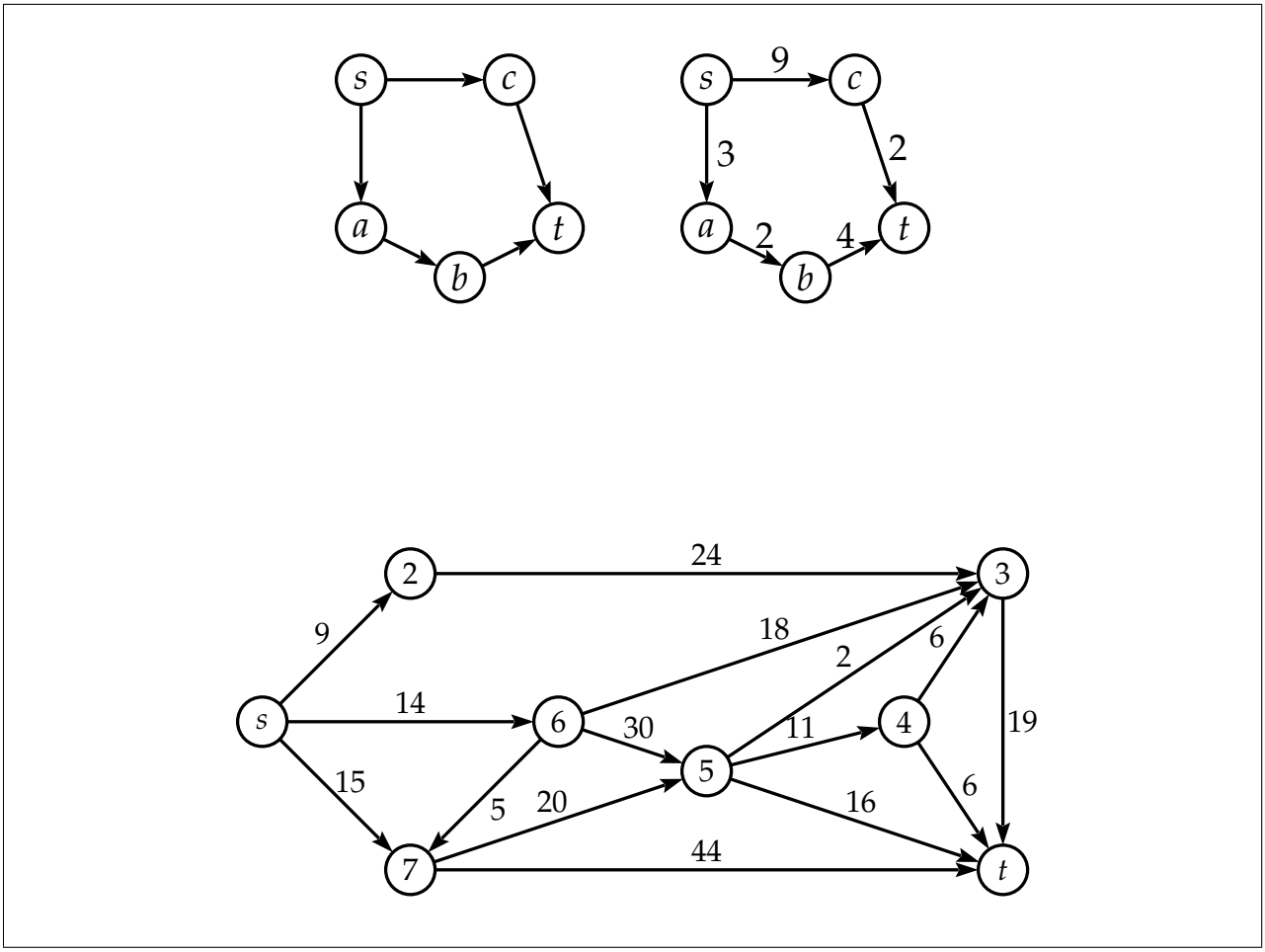
Algorithms: Introduction to Dijkstra's Algorithm

Model 1: Some graphs



(Review) Begin by considering Graph 1 (top left).

1 What is the shortest path from s to t? How long is it?

2 What algorithm could you use to find it?

Now consider Graph 2 (top right). It looks just like Graph 1 except that each edge is labelled with a *weight*.

3 What do you think the "length" of a path means in this graph?

4 According to your definition of length, what is the shortest path from s to t? How long is it?

5 Why wouldn't your answer to Question 2 work here?

Now consider Graph 3. Imagine that each edge is a pipe that only allows water through in the direction the arrow is pointing. The number on the edge indicates how many seconds it takes for water to flow from one end of the pipe to the other. Now imagine that we hook up an (infinite) source of water to vertex s and watch the water start flowing through the network of pipes. Of course, water always flows in all possible directions. For example, as soon as we hook up the water source to vertex s, water immediately begins flowing along all three pipes leaving from s.

6 What is the first vertex (besides s) the water reaches? At what time does this happen (counting in seconds from the moment we hook up the water to vertex s)?

7 What is the second vertex the water reaches? At what time does it happen?

8 What is the third vertex the water reaches? At what time?

9 Suppose we changed the length of the pipe $s \to 7$ from 15 to 20. How (if at all) would this change your answers to the previous questions?

10 Draw the situation at 32 seconds. (You may wish to draw directly on the graph in the model; or you can make a separate copy.) Which vertices has the water reached? Which pipes are full?

11 After 32 seconds, which new vertex will the water reach next?

12 At what time does the water first reach vertex t?

13 What is the length of a shortest path from s to t?

14 How does thinking about water flooding the graph help us solve the problem of finding shortest paths? Make a conjecture relating water in the graph to shortest paths between s and other vertices.

This process of keeping track of water flooding a graph from a start vertex and seeing when it first reaches each other vertex is known as *Dijkstra's algorithm*.

15 What similarities or differences do you see between BFS and Dijkstra's algorithm?

16 If you had only a black box that could run Dijkstra's algorithm, and someone gave you an *unweighted*, directed graph and asked for the shortest path between two vertices, what would you do?

17 How about vice versa? That is, imagine you have only a black box that can run BFS, and someone gives you a directed graph with positive integer weights on the edges, and asks for the shortest path between two vertices. What should you do?

18 Does Dijkstra's algorithm work if there are edges with negative weight? Explain why it works, or draw an example graph to illustrate why it doesn't.