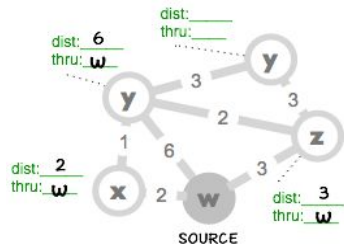


Activity Guide - Dijkstra's Shortest Path Algorithm



You be the computer!

Below is an adaptation of a famous algorithm to find the shortest path from one node to *all the other nodes* in the graph. It is known as **Dijkstra's algorithm**, after its inventor, Edsger Dijkstra ("Dike-strah"). Today you will act as the computer, interpreting the instructions for the algorithm to produce a **shortest path spanning tree** rooted at your router.



Here's what Dijkstra's algorithm does, in informal English:

The algorithm has you systematically find the shortest path from a **source node** to every other node in the graph by considering one node at a time. Along the way, you keep track of the total distance from the source going through other nodes and update those distances when you find shorter paths. The algorithm below is also written in English but is a bit more structured than usual; it's closer to a real computer program.

Directions:

With a partner, try to follow the algorithm below and track your progress on the graph provided to you. Follow the steps in order, starting at step 0.

Algorithm: Dijkstra's Single Source Shortest Path

Setup: On the graph, your source node is labeled **Source** and is shaded in. For each of the neighbors of your **Source** node, we've set the initial values for **dist** and **thru**.

NOTE: dist will be used to keep track of the total best distance from Source

1	<p>Identify the unshaded node with the smallest dist value (ignore blanks). If there is a tie for node with smallest dist value just pick either one.</p> <p>Label this node Current.</p> <p><i>In the example on the right, current becomes x, because it is the node with the smallest dist of the three options → x = 2, z = 3, y = 6 and v is blank</i></p>	
2	<p>For each unshaded Neighbor of Current complete step 3. (A Neighbor is a node with a direct link to Current.)</p>	
3	<p><u>Compute distance:</u> Add Current's dist + weight of edge to Neighbor.</p> <p>Check if this distance is less than Neighbor's current dist <u>OR</u> Neighbor's dist is blank. If it is less (or blank)</p> <ol style="list-style-type: none"> update Neighbor's dist to be this new dist update Neighbor's thru to be the name of the Current node. <p>Repeat process for all the neighbors of Current.</p>	
4	<p>Once done visiting each neighbor of Current...</p> <ol style="list-style-type: none"> Shade in Current node Cross out the Current label. (You're now done with this node) 	
5	<p>If all nodes in the graph are shaded → go to step 6 otherwise → go back to step 1</p>	
6	<p>For each node in the graph: highlight the edge between that node and its thru node.</p>	
7	<p>The shortest path tree from the source is now highlighted. Yay!</p>	