The shortest-path problem
David Gries

We have a graph with n nodes for some n > 0. The nodes are given distinct numbers, in green below, in the range 0..n-1. The edges could be directed or undirected.

Each edge has a positive weight, written in red below. If the nodes were cities and the edges roads, the weights could be the distance between the cities, or the estimated driving time, or something similar. We use positive integers for edge weights.

The notation

wgt(v1, v2)

denotes the weight on the edge from node v1 to node v2. For example, in the graph in the window, wgt(0, 2) is 4 and wgt(2, 3) is 1.

The *distance* of a path is the sum of the weights of the edges on the path.

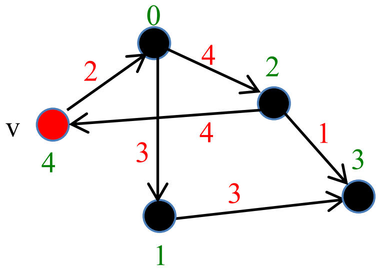
Node v is designated as the start node. In this case, v is the red node on the left, so v is 4.

The purpose of the shortest-path algorithm is to calculate the distance of the shortest path from v to each of the reachable nodes of the graph. For example, the distance of the shortest path:

from v to v is 0,
from v to 2 is 6 (there is only one path), and
from v to 3 is 7 (there are two paths).

We'll store the distances in an array d[0..n-1] —d standing for *distance*. So, upon termination, for each node w, d[w] will contain the distance along the shortest path from v to w,

To the right is the resulting array d for our example graph.



d[0] = 2
d[1] = 5
d[2] = 6
d[3] = 7
d[4] = 0