Simulating the DV Algorithm

In this lab you will write the code for the Distance Vector algorithm. The algorithm for this is shown on page 383 of your textbook; however, you will be adding additional steps to keep track of the actual routing path. You will also be implementing a slightly simpler version of the algorithm in which we deal with updates from a single neighbor at a time, rather than all at once.

The algorithm shows how router $x$ updates its own distance vector $D_x$ based upon the reception of distance vector $D_y$ from another router, $y$. You will be adding one additional component to this, namely the updating of a “next” vector that states which is the next node to visit along a shortest path. For example, in the following graph, the shortest path from 2 to 4 has length 8 and goes through node 3; therefore, for router number 2, $D_2(4) = 8$ and $next_2(4) = 3$.

Here is the revised algorithm:

**Initialization:**
...will be provided for you in the code ...

**Loop:**
wait until I receive a distance vector $D_w$ from some neighbor $w$
for each node $y = 0, 1, \ldots, n-1$:
  change = false
  if $D_x(y) > cost(x, w) + D_w(y)$ then
    change = true
    update $D_x(y)$ and $next_x(y)$
  if change
    send $D_x$ to all neighbors
You will be given several files, of which only the `Router.java` file needs to be modified. However, you might benefit from looking at the other files (for instance, to see what the methods are for the various classes like `NbrCostPair` or `Neighbor`).