Another shortest path (Bellman-Ford).

Dijkstra's algorithm is good if there is no edge of negative length (Exercise?), but
Bellman-Ford works as long as there is no negative cycle, i.e., a cycle whose edges
sum to a negative value and it can detect such cycles.

Algorithm Bellman-Ford(G, V);
begin
SP[V] = 0; SP[W] = ∞ for W ≠ V;
for i = 1 to |V| - 1
  for each (u, w) ∈ E  // Relaxing edge (u, w)
    if SP[u] + length(u, w) < SP[w] then
      SP[w] = SP[u] + length(u, w)
end;

Proof by Induction:

III: If there is a path from u to v with at most i edges, then SP[v] is at most the
length of the shortest path from v to u with at most i edges, where i is the repetitions.

II: Consider the shortest path P from v to u with at most i edges. Let w be the last vertex
before u on this path. Then the path from v to w is a shortest path from v to w
with at most i-1 edges and by induction SP[w] after i iterations is at most the length
of this path. Thus SP[w] + length(w, u) is at most the length of P and we find it in
the i-th iteration.

Now if there is no negative cycle, the length of any shortest path in terms of
edges is at most |V| - 1 and thus we find it by III for i = |V| - 1.

It itself is a good property of this algorithm, which has applications in routing
protocols as well.
All-pairs shortest paths problem (Floyd-Warshall algorithm)

The problem: Given a weighted graph $G = (V,E)$ with non-negative edge lengths, find the minimum-length paths between all pairs of vertices.

Of course, we can run $|V|$ times Dijkstra with total time $O(|V||V| + |E| \log |V|) = O(|V|^3 \log |V|)$, which is good for sparse graphs but not the best for dense graphs.

Algorithm Floyd-Warshall(G):

begin
  for $m = 1$ to $n$ do // induction sequence: loop
    for $x = 1$ to $n$ do
      for $y = 1$ to $n$ do
        if weight$[x,m] + \text{weight}[m,y] < \text{weight}[x,y]$ then
          weight$[x,y] = \text{weight}[x,m] + \text{weight}[m,y]$

Proof by Induction:

I.H.: We know the lengths of the shortest paths between all pairs of vertices such that only $K$-paths, i.e., except endpoints, the highest-labeled vertex on the path is labeled $K$, for some $K \leq m$ are considered. In $ith$ iteration of loop, we computed all these.

B.F.: for $m=1$, the basis is correct since we have only direct edges as paths.

For $m$, the shortest path between any pair can have $V_m$ at most one and then the paths to and from $V_m$ are $K$-paths, for $K < m-1$. Since we sum the paths to and from $V_m$ and compare it with the best path found so far, we are done.

As you saw in both Bellman-Ford and Floyd-Warshall, the whole idea is to get I.H. correct and the rest is trivial.