Another single-source 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 detects 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;
```

Time complexity: clearly it is $O(|V||E|)$ which is worth than Dijkstra's $O((|V|+|E|)\log|V|)$

Proof is by Induction:

<u>IH</u>: If there is a path from v to u with at most i edges, then sp[u] is at most the length of the shortest path from v to u with at most i edges, <u>where i is the repetitions</u>.

<u>Pf</u>: 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 part of P 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 iteration 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 IH for $i=|V|-1$.

IH itself is a good property of this algorithm, which has applications in routing protocols as well.

All-pairs shortest paths problem (Floyd-Warshall algorithm)

<u>The problem</u>: 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 $O(|V|(|V|+|E|)\log|V|)=O(|V||E|\log|V|)$ 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]+weight[m,y] < weight[x,y] then
            weight[x,y] = weight[x,m]+ weight[m,y];
```

Time complexity: Very simple algorithm to implement in $O(|V|^3)$.

Proof by Induction:

IH: We know the lengths of the shortest paths between all pairs of vertices such that only k-paths, i.e., except end-points, the highest-labeled vertex on the path is labeled k, for some $k \leq m$, are considered. In ith iteration of <u>loop</u>, we computed all these.

<u>Pf</u>: 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 $k \leq m-1$. Since we sum the paths to and from V_m and compare it with the best path found so far, in the algorithm, we are done. ∎

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