The All Pairs Minimum Path Problem

We are given a weighted directed graph $G = (V, E, W)$. That is, if $e = (u, v) \in E$, then $W(e) = W(u, v)$ is the *weight* of the edge e. A solution to the problem consists of arrays $\{V[u, v]\}_{u,v \in V}$ and $\{back[u, v]\}_{u,v \in V}$, such that $V[u, v]$ is the minimum weight of any path from u to v, while $back[u, v]$ is the next-to-the-last vertex of one of those minimum paths. There is no solution if G has a negative cycle. For convenience, we assume that the vertices of G are the integers $1, \ldots n$.

The Floyd–Warshall Algorithm

```
for all 1 <= i <= n
   V[i,i] := 0
   back[i,i] := * \\ undefined
endfor
for all 1 <= i <= n
   for all 1 <= j <= n
      if there is an edge from i to j
         back[i,j] := i
         V[i,j] := W(i,j)
      else
         back[i,j] := * \\ undefined
         V[i,j] := infinity
      endif
   endfor
endfor
for all 1 <= j <= n
   for all 1 <= i <= n
      for all 1 <= k <= n
         temp := V[i,j] + V[j,k]
         if temp < V[i,k]
            V[i,k] := temp
            back[i,k] := back[j,k]
         endif
      endfor
   endfor
endfor
```