1 FW All-Pairs SPs

1.1 Data structure

`d(i,j,k) =`

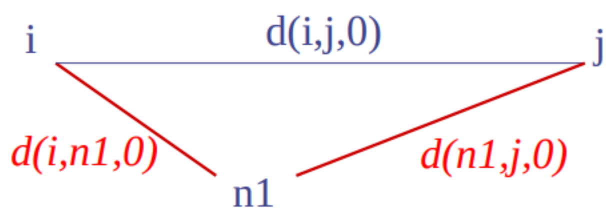
- shortest distance between nodes i and j
- using only the nodes 1,,k as potential allowed intermediary points

1.2 Initialisation of data structure

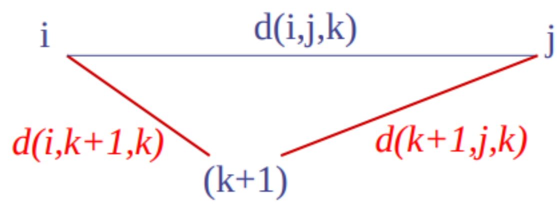
`d(i,j,0) =` best distance between nodes i and j, but not using any intermediate nodes, so only using a single edge, hence `d(i,j,0) = w(i,j)` if there is an edge i to j, otherwise `Inf`

1.3 All-Pairs SPs

- Now suppose that we add the node n1 to the set of nodes that can be intermediates, i.e. consider k = 1
- Best path is now the best of either direct, or via n1.
- `d(i,j,1)`
  = `min ( d(i,j,0), w(i,n1) + w(n1,j) )`
  = `min ( d(i,j,0), d(i,n1,0) + d(n1,j,0) )`



- Now suppose that we add the new node (k+1) to the set of via nodes that can be intermediates, but have already considered k of them
- Best path is now either direct using only the k via nodes already accounted for, or else also via node k+1 (and using the previous k vias)
- `d(i,j,k+1) = min ( d(i,j,k), d(i,k+1,k) + d(k+1,j,k) )`



1.4 Equations

- `d(i,j,0)`
  = `w(i,j)` if there is an edge i to j
  = `Inf` otherwise
- `d(i,j,k+1) = min ( d(i,j,k), d(i,k+1,k) + d(k+1,j,k) )`
- `d( i, i ) = 0 forall i`

1.5 Complexity

Because we have 3 variables, i,j,k we will need 3 levels nested for loop, where `i = j = k = |V|`, so worst case is $O(|V|^3)$

1.6 Digraphs (Directional graphs)

The algorithm also works on directional graphs.The initial matrix `d(i,j,0)` need not be symmetric, but then the remaining calculations use exactly the same formulas

1.7 Negative edges

FW even works if some (directed) edge weights are negative

- BUT it is essential that there are **no cycles of total negative weight**
- Otherwise simply repeatedly following around the negative cycle may reduce lengths to be as negative as desired, so there is **no shortest path**