Algorithm: ComputeFlow

Input: an undirected graph g = (V,E), a tuple of node to other nodes distance, a tuple of node’s number of shortest paths.

Output: a tuple representation for the flow between Edges E

1. Initialize an empty array edgeflows
2. Value 🡨 0
3. Foreach i,j in E:
   1. Edgeflows[I,j] 🡨 0
4. Maxdistance 🡨 Max |nodedistance|
5. While maxdistance > 0 do:
   1. If maxdistance = Null:
      1. Break
   2. For I in nodedistance do:
      1. If nodedistance[i] = maxdistance:
         1. Foreach j in g[i] if nodedistance[j] > nodedistance[i]:
            1. Value 🡨 value + edgeflow[j,i]
      2. Computedflow 🡨 (1+value)/numpath[i]
      3. For node in g[i] do
         1. If nodedistance[node] < nodedistance[i]:
            1. Edgeflows[node,i] 🡨 (computedflow)(numpath[node]
   3. Maxdistance 🡨 maxdistance – 1
6. Return edgeflows