Algorithm: MatrixBased\_DNs

Input: a graph g = (V,E) and a starting node s an element of V

Output: an array that stores the distance from node s and other nodes in V

1. Initialize an empty array distancevalues
2. Initialize an empty array searchcommunity
3. Distancevaluess 🡨 0
4. R 🡨 1
5. Initialize a |V| x |V| matrix distancematrix
6. Community 🡨 ConnectedComponents(g) \\Find the community that node s is within
7. Foreach subset in community do:
   1. If node s in subset do:
      1. Searchcommunity union subset
   2. Else
      1. Foreach node in subset do:
         1. Distancevaluesnode = Null
8. Findnumbernodes 🡨 |searchcommunity|
9. While Findnumbernodes is not zero do:
   1. Power 🡨 ComputeNumPath(g,r)
   2. Raise distancematrix to r
   3. Foreach j in distancematrix[s,j] do:
      1. If distancematrix[s,j] > 0 and distancevaluesj = 0:
         1. Distancevaluesj 🡨 r
   4. Findnumbernodes 🡨 findnumbernodes – power
   5. R 🡨 r + 1
10. Return distancevalues