

Annexe 1

Input:

- A weighted (possibly directed) graph G containing n nodes.
- The $n \times n$ cost matrix \mathbf{C} associated to G , containing nonnegative edge costs.

Output:

- The $n \times 1$ column vector **bet** containing the betweenness scores.

```

1. bet  $\leftarrow$  0
2. for  $i = 1$  to  $n$  do      > do the following for each node  $i$  regarded as the origin
3.    $Pred_i^* \leftarrow \emptyset$  for all  $j = 1, \dots, n$   > set of predecessors on shortest paths from  $i$  to  $j$ 
4.    $\sigma \leftarrow \mathbf{0}$ ;  $\sigma_i \leftarrow 1$   >  $\sigma_j$  = number of shortest paths from  $i$  to  $j$  found so far
5.    $\Delta \leftarrow \infty$ ;  $\Delta_i \leftarrow 0$   >  $\Delta_j$  = minimum path cost from  $i$  to  $j$  found so far
6.    $S \leftarrow$  empty stack
7.    $Q \leftarrow \{i\}$       >  $Q$  is usually implemented as a min-priority queue
8.   while  $Q \neq \emptyset$  do      > solving the single-source shortest-path problem
9.      $j \leftarrow \arg \min_{k \in Q} \{ \Delta_k \}$ ;  $Q \leftarrow Q \setminus \{j\}$   > find the closest node in  $Q$  and remove it from  $Q$ 
10.    push  $j$  into  $S$ 
11.    for all  $k \in Succ(j)$  do  > loop on successors of node  $j$ 
12.      if  $\Delta_k = \infty$  then  >  $k$  is not yet in  $Q$ 
13.         $\Delta_k \leftarrow \Delta_j + c_{jk}$ 
14.         $Q \leftarrow Q \cup \{k\}$   > integrate  $k$  in  $Q$ 
15.      else if  $\Delta_j + c_{jk} < \Delta_k$  then  > a better path to  $k$  is found
16.         $\Delta_k \leftarrow \Delta_j + c_{jk}$ 
17.         $\sigma_k \leftarrow 0$ ;  $Pred_k^* \leftarrow \emptyset$   > reset  $\sigma_k$  and  $Pred_k^*$ 
18.      end if
19.      if  $\Delta_k = \Delta_j + c_{jk}$  then  >  $j$  lies on a shortest path to  $k$ 
20.         $\sigma_k \leftarrow \sigma_k + \sigma_j$   > accumulate the number of paths
21.         $Pred_k^* \leftarrow Pred_k^* \cup \{j\}$   > collect predecessors of  $k$  on all shortest
paths
22.      end if
23.    end for
24.  end while
25.  dep  $\leftarrow \mathbf{0}$ 
26.  while  $S$  is not empty do > accumulating the contributions to betweenness
27.    pop  $k$  from  $S$   > traverse node  $k$  in decreasing order of the path cost from  $i$ 
28.    for each  $j \in Pred_k^* \setminus \{i\}$  do  > for each edge  $(j,k) \in P_{ik}^*$ 
29.       $dep_j \leftarrow dep_j + \frac{\sigma_j}{\sigma_k} (1 + dep_k)$ 
30.    end for
31.     $bet_k \leftarrow bet_k + dep_k$ 
32.  end while
33. end for
34. if  $G$  is an undirected graph then  $bet \leftarrow \frac{1}{2}bet$ 
35. return bet

```