

Algorithm 1: All Simple Paths

KCores(G)

Input: $G = (V, E)$, a graph

Data: *visited*, an array of size $|V|$

path, a stack of nodes

paths, a stack containing stacks of nodes

Result: *paths* contains all paths from s to t

```
1   $mh \leftarrow \text{CreatePQ}()$ 
2   $d \leftarrow \text{CreateArray}(|V|)$ 
3   $p \leftarrow \text{CreateArray}(|V|)$ 
4   $core \leftarrow \text{CreateArray}(|V|)$ 
5  for  $v = 0$  to  $|V|$  do
6       $d[v] \leftarrow |\text{AdjacencyList}(G, v)|$ 
7       $p[v] \leftarrow d[v]$ 
8       $core \leftarrow 0$ 
9       $dn \leftarrow \text{CreateArray}(2)$ 
10      $dn[0] \leftarrow p[v]$ 
11      $dn[1] \leftarrow v$ 
12      $\text{InsertInPQ}(mh, dn)$ 
13 while  $\text{SizePQ}(mh) > 0$  do
14      $t \leftarrow \text{ExtractMinFromPQ}(mh)$ 
15      $core[t[1]] \leftarrow p[t[1]]$ 
16     if  $\text{SizePQ}(mh) = 0$  then
17         return  $core$ 
18     foreach  $v$  in  $\text{AdjacencyList}(t[1])$  do
19          $d[v] \leftarrow d[v] - 1$ 
20          $odn \leftarrow [p[v], v]$ 
21          $p[v] \leftarrow \text{Max}(core[t[1]], v)$ 
22          $ndn \leftarrow [p[v], v]$ 
23          $\text{UpdatePQ}(mh, odn, ndn)$ 
24 return  $core$ 
```