Algorithm 1 Disjoint communities formation

Require: $iters$, $neighbor\_selection\_threshold$, G
  $edges \leftarrow$ number of edges in G
  $num\_iterations \leftarrow iters * edges$
  while $num\_iterations$ - - **do**
    e $\leftarrow$ edge with highest edge betweenness centrality
    G $\leftarrow$ G - e
  end while
  $disjoint\_communities \leftarrow$ connected components of G

Algorithm 2 Conversion of disjoint communities to overlapping communities

Require: *disjoint_communities*

```
communities ← [ ]
for C in disjoint_communities do
    temp_list ← [ ]
    for n in C do
        temp_list ← temp_list + n
        for nn in neighbors of n do
            if nn in C then
                continue
            end if
            total_neighbor_count ← number of neighbors of n
            in_c_count ← 0
            for nnn in neighbors of nn do
                if nnn in C then
                    in_c_count ← in_c_count + 1
                end if
            end for
            if in_c_count / total_neighbor_count ≥ threshold then
                temp_list ← temp_list + nn
            end if
            communities ← communities + temp_list
        end for
    end for
end for
```