

NETWORKS AND COMPLEXITY

Solution 2-10

*This is an example solution from the forthcoming book *Networks and Complexity*.*

Find more exercises at <https://github.com/NC-Book/NCB>

Ex 2.10: Component algorithm [4]

Invent an algorithm for finding all the nodes that are in the same component as a given node.

Solution

There are many possible algorithms. The idea of the solution below is to keep a list of nodes that we know to be in the component. For each node on the list we find all of its neighbors that are not on the list yet and add them to the list. To do this efficiently the list is actually split into two sets, the set C for which we have already checked the neighbours and the set A for which we have yet to check the neighbours.

1. Let x be the starting node, A a set containing x and C an empty set.
2. Add to A all neighbours of x that are not in A or C .
3. Remove x from A and add it to C .
4. If A is an empty set terminate, else pick an element from A , let this element be the new x and go to 2.

When the algorithm terminates all nodes in the component of the starting node will be in C .

[One can also answer this question using a cheap trick: 1. Consider all links to have the length one. 2. Run Dijkstra's algorithm from the starting node. 3. The set of nodes that are in the same component as the starting node is the set of nodes that are less than infinitely far away from the starting node.]