Fundamental Algorithms 11

Exercise 1 (Modified Graph Traversal)

Consider the modified traversal algorithm for graphs and trees ModTrav.

Algorithm 1: ModTrav

Input: V: Node
act ← []; // Local queue of active nodes
for (V, W) ∈ V.edges do
    if mark[W.key] = 0 then
        visit(W);
        mark[W.key] ← 1;
        act ← act ◦ W;
    end
end
for W ∈ act do ModTrav(W);

1. Given the graph above, in which order the function visit is called on the nodes by this algorithm? Number the nodes accordingly. It is given that, initially, the start node S is marked and the rest of the nodes are unmarked. It is not specified in which order edges outgoing from a node V are stored in the list V.edges – you may assume any order you like.

2. In the same graph, mark the edges that are part of the spanning tree computed by the algorithm.

3. Now assume that the second for-loop is changed into a parallel loop. Discuss whether there can be concurrent read or write access to the elements of the array mark. Think about what happens if the graph is a tree.