1 BFS

Do Exercise 22.2-6. You do not have to give a rigorous proof of the correctness of your algorithm, but you do need to give an intuitive explanation why your algorithm works. Remember to analyze the running time.

2 Topological sort

Do Exercise 22.4-2 (on p. 552).

3 Querying on a tree

You are given a rooted binary tree \( T = (V, E) \), along with a designated root node \( r \in V \). The size of \( V \) is \( n \). Recall that a node in a tree keeps tracks of its descendants and its parent node. Also recall that node \( u \) is said to be an ancestor of node \( v \) in the rooted tree, if the path from \( r \) to \( v \) in \( T \) passes through \( u \).

A commonly performed querying on the trees is: given two nodes \( u \) and \( v \), is \( u \) an ancestor of \( v \)? You wish to preprocess the tree so that queries of this form for any two nodes \( u \) and \( v \) can be answered in constant time. The preprocessing itself should take linear time. That is, you can spend \( O(n) \) time before any query arrives; then you must answer each query like “is node \( u \) an ancestor of node \( v \)?” in constant time.