

Exercises on Binary Tree network

Algorithmic Data Mining

Advanced Algorithms

E.1 Check if Sorted

Even though the binary tree network is not really suited for sorting algorithms, it can still be used to reasonably quickly check whether some input sequence x_0, x_1, \dots, x_{N-1} is already sorted or not. To be sorted requires that $x_i \leq x_{i+1}$ for every $0 \leq i \leq N - 2$. Let the N inputs be initially input one element in each leaf of a binary tree with $P = 2N - 1$ processing units with constant $O(1)$ memory, design a parallel algorithm for the network that outputs (in the root) *True* (1) if the input sequence is already sorted, and *False* (0) otherwise.

E.2 Bits Sorting

A special case when the binary tree network becomes suited for sorting is when the input sequence b_0, b_1, \dots, b_{N-1} consists solely of 0s and 1s. As usual, the N inputs will be initially placed in the N leaves of a binary tree with $P = 2N - 1$ total processing units with constant $O(1)$ memory. Leverage the peculiarities of the input to design an efficient parallel algorithm to sort the values in the leaves so that in the end they are sorted from left to right.