

Tutorial 6 for COMP 526 – Efficient Algorithmics, Fall 2022

Problem 1 (Parallel And)

We consider the problem of computing the logical *and* (\wedge) of an array $B[0..n-1]$ of n Boolean values (n bits), i.e., the result should be *true* if and only if all n entries are true. (We assume here that each bit is stored as a full word.)

- Design a CREW-PRAM parallel algorithm for computing the “logical and” of $B[0..n-1]$. Your algorithm should have $\mathcal{O}(\log n)$ time (span) and $\mathcal{O}(n \log n)$ work.
- Can you make the algorithm work-efficient?
- Now consider a CRCW-PRAM; you can choose a write-conflict resolution rule that is convenient for your purposes. Design a *constant-time* parallel algorithm for computing the logical and.

Problem 2 (Suffix trees and friends – Part I)

Consider the text $T = \text{abbabb}\$$.

What is n here? (exactly follow the convention from the lecture!)

Construct/Draw the

- standard (not compacted) trie of all suffixes of T ,
- suffix tree of T (human version) with string labels on edges and leaves,
- suffix tree of T (computer version) as it is stored, i.e., offsets in nodes, starting index in leaves, first characters on edges,