

Exercise 4 - PRAM Algorithms

- 4-1 (i) **Finding the first one:** Given an array $A[1 \dots n]$ of boolean elements, the problem of finding the first one is to find an index l such that $A[l] = 1$ and $\forall j < l, A[j] = 0$. Give a $O(n/\log n)$ -processor EREW PRAM algorithm that finds the first one in $O(\log n)$ time.
- (ii) **Array Compaction:** Given an array $A[1 \dots n]$, compute an array B such that $B[i]$ contains the i -th non-zero element of A . Show that it can be done using a $O(n/\log n)$ -processor EREW PRAM algorithm that runs in $O(\log n)$ time.
- 4-2 **Sorting Integers:** Let $A[1 \dots n]$ be an array with $A[i] \in \{1, \dots, \sqrt{n}\}$ such that for all $1 \leq k \leq \sqrt{n}$, the frequency of k (number of indices such that $A[i] = k$) is bounded by $O(\sqrt{n} \log n)$. Give a $O(n)$ -processor PRAM algorithm that sorts A in $O(\log n)$ time with high probability.