Fundamental Algorithms 10

Exercise 1 (Parallel sort)

BUCKETSortPRAM is a proposed parallel implementation of the BUCKETSort algorithm in a PRAM, n processor model. concat is a sequential algorithm that concatenates the given buckets in the provided order. The function index distributes the array elements evenly into buckets, i.e. elements are assigned to buckets with (roughly) the same probability.

Algorithm 1: BucketSortPRAM

Input: $A$: Array[1..n]
Result: Array with contents of $A$ sorted

$B \leftarrow$ Array[1..n]:
for $i = 1$ to $n$ in parallel do
  insert($B[index(A[i])]$, $A[i]$);
for $i = 1$ to $n$ in parallel do
  BubbleSort($B[i]$);
return concat($B$);

1. For the two parallel loops in BucketSortPRAM, state for both arrays $A$ and $B$ whether there is concurrent or exclusive read / write access to their elements.

2. Describe a parallel algorithm, for an EREW PRAM model, which concatenates the buckets in the last step, using as many processors as possible. What is the parallel complexity to concatenate all buckets (depending on $n$), and how many processors can your algorithm use?