

COL380 REVIEW QUESTIONS

(Parallel Algorithms)

1. Provide an efficient implementation of prefix sum for GPUs.
2. Provide an efficient implementation of Prefix sum on multiple machine using MPI.
3. A collision detection algorithm tests for intersection between every pair of elements comprising a body. Each such test takes $O(1)$ time sequentially. Provide an efficient EREW PRAM algorithm for performing all tests given a body with n elements.
4. In question 3, the vast majority of pairs are expected to NOT intersect. To make the testing efficient, the elements are divided into m sets (each with an arbitrary number of elements), $m \ll n$, and an oracle is provided which, given two sets A_0 and A_1 can predict in $O(1)$ sequential time if any elements of A_0 could possibly intersect with any element of A_1 (it does not tell which). Given sets $A_0..A_{m-1}$, with the count of elements in set $A_i = n_i$, provide an efficient EREW PRAM algorithm to compute the intersecting element pairs. Note that if the oracle cannot eliminate pair $A_i A_j$, every element of A_i must be tested against every element of A_j . Is the efficiency of CREW PRAM algorithm lower?
5. Provide an $O(\log n)$ time work-optimal EREW PRAM algorithm to find the lowest index i , such that $A[i] == 1$, given array A with n boolean values.
6. Provide an $O(1)$ time work-optimal common CRCW PRAM algorithm to find the lowest index i , such that $A[i] == 1$, given array A with n boolean values.
7. Show that any EREW PRAM algorithm that uses $P(n)$ processors on input of size n can be implemented on a machine with only $\max(n, P(n))$ shared memory locations (assume that each processor has unlimited local memory and the initial input is in the shared memory).
8. Show that the LINK-UNLINK-based parallel priority queue retains its property (i.e., intra-tree key order, 1-3 roots per rank, and inter-root key order) after each valid operation.
9. Provide a GPU-friendly PRAM algorithm to perform radix sort.
10. In a graph, $N[i][\]$ is an array of indexes of neighbors of node i . (Let $n[i]$ denote the number of neighbors of node i , and nn denote the total number of nodes.) Provide a common CRCW PRAM algorithm to perform a BFS traversal of this graph starting at index s . Analyze its time and work complexity.
11. Provide an CREW PRAM BFS algorithm (for the graph in the previous question). Analyze its time and work complexity.