Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chris Pinkau

Winter Semester 2014/15
Problem Set 04
November 3rd, 2014

Parallel Algorithms

Due date: November 10th, 2014 **before** *class!*

Problem 1 (10 Points)

Show how to solve the parallel search problem on a sorted array of n elements in $\mathcal{O}(\log n - \log p)$ steps on an EREW PRAM with p processors, provided that the search key can be accessed concurrently by all the processors.

Problem 2 (10 Points)

Suppose that we use the $\mathcal{O}(\log^* n)$-time 3-coloring algorithm (Lemma 12 from the lecture) to identify a large independent set needed in the List Ranking algorithm. Can you make the resulting list-ranking algorithm run in $\mathcal{O}(\log n)$ time with $\mathcal{O}(n \log^* n)$ operations? Justify your answer.

Problem 3 (10 Points)

Suppose a tree T is given by its adjacency lists without the additional pointers. Develop an algorithm to determine the Euler tour. What are the corresponding bounds for the running time and the total number of operations? Can you improve the performance of your algorithm if the degree of each vertex is bounded by a constant?

Problem 4 (10 Points)

Consider an arbitrary rooted tree $T = (V, E)$ such that for each vertex v you are given the next sibling of v, denoted by $s(v)$, and the first child of v, denoted by $fc(v)$ (if no sibling exists, then $s(v) = 0$; if v is a leaf, then $fc(v) = 0$).

1. Develop an $\mathcal{O}(\log n)$ time algorithm to identify, for each vertex v, the parent $p(v)$ of v. Your algorithm must use a linear number of operations.

2. Develop an $\mathcal{O}(\log n)$ time algorithm that stores the leaves in consecutive memory locations as they appear in T from left to right. Your algorithm must use a linear number of operations.