Computational Geometry: Theory and Experimentation (2023)
PRAM Exercises

Peyman Afshani

Focus:

Implementing Simple Parallel Algorithms in Java
Solving Theoretical Exercises in PRAm

A General Hint: Combining Known Techniques

Elaboration: Relying on newly or previously acquired techniques is an excellent method of solving difficult problems.

1 Parallel Programming in Java

Download `parallelSum.zip` from here. Note that you probably need to be logged into brightspace for the link to work. The zip file contains two Java programs. Both of them try to create a lot of threads to sum up the numbers in an array. The `ExpParallel.java` creates a fixed number of threads to sum up the numbers in the array. The other program is using the *fork-join* framework to do the same thing. The programs are included to give you a starting point for writing parallel programs in Java. You should already be a bit familiar with this task but as with most programming tasks, try to read the relevant documentation in Java API.

Task 1. Modify any of the above programs to find the minimum or the maximum element in the array.

2 A Theoretical Question

Task 2. Show that in the common CRCW PRAM model, it is possible to find the minimum element in an array of size n in $O(\log\log n)$ time, using $O(n)$ processors.

Task 3. Show that in the common CRCW PRAM model, it is possible to find the minimum element in an array of size n in $O(\log\log n)$ time, using $O\left(\frac{n}{\log\log n}\right)$ processors.

Remarks. The above result is *optimal*. Note that the work is optimal since it is $O(n)$, however, there are also known lower bounds that show in the common CRCW model, it is not possible to reduce the parallel time below $\log\log n$.