

Parallel Haskell – Homework

Marek Imiełowski

Your task in the exercises below is to make a parallel implementations of the given algorithms. You are allowed to use `Eval` (with or without strategies) and `Par` monads – grading will depend solely on code clarity and performance. As these exercises are centered on achieving the best parallel speedup you can, keep your implementations simple and focus on getting parallelism right, not on the overall performance. Feel free to base your solutions on some freely available sequential implementations, eg. from *Rosetta Code*.

Ex 1. (3 pts.)

Create a parallel implementation of quick sort for standard Haskell lists.

Ex 2. (6 pts.)

Parallelize the library function `scanl1`.

Ex 3. (6 pts.)

Create a parallel implementation of the Cooley-Tukey fast fourier transform algorithm on standard lists of complex values.

The deadline is **25th June**. Good luck!