# Design & Analysis of Parallel Algorithms: Exercise Sheet 1

*Please be sure that you have read, understood and adhered to the School and University guidelines on late submission of coursework and academic misconduct, which can be found via the course webpage.*
*This sheet accounts for 10% of the course final mark. You should submit your work electronically, in pdf, by* **4pm on Thursday 24th October 2013**, *using the DICE command*

```
submit dapa 1 dapaEx1.pdf
```

*If you are* **not** *a School of Informatics student then you may submit your work by e-mail to* `mic@inf.ed.ac.uk`. *The best marks will be awarded for simple, correct and clearly argued content. It is important that you use the filename given, and that your document is in pdf.*

1. (Rationale: This question is designed to test your understanding of the basic PRAM machine models, and of the related concepts of cost optimality and scalability.)

   Course overhead slide 15 sketched an $n^2$ processor, constant time sorting algorithm for the CRCW PRAM variant which assumes addition to resolve clashing writes. We have already noted that this is not cost optimal.

   (a) Explain how the algorithm can be scaled down to operate upon an $n$ processor EREW PRAM, so that the first phase (computing the ranks) runs in $\Theta(n)$ time and the second phase (rearranging the data) runs in constant time. Your solution should not expand the amount of memory required by more than a constant factor. For example, using twice as much memory is OK, but $n$ times as much is not. (NB you are being asked to scale the algorithm down, not to replace it with a completely different algorithm).

   (b) Using any results which seem relevant to you, discuss the chance of obtaining a cost optimal algorithm by such a scaling down. As above, you are being asked to think about *scaling down* the given algorithm, **not** replacing it with some new algorithm.

   (40 *marks*)

2. (Rationale: This question is designed to test your understanding of reduction and prefix operations, and to think creatively about algorithmic problem solving. It also further tests your understanding of the concepts of cost optimality and scalability.)

   Design an algorithm for the $n$ processor CREW PRAM which solves the following problem: given an array of $n$ integers, re-arrange the array so that those integers which are smaller than the average (of all the integers) precede those which are equal to or greater than the average, while preserving the original ordering within each of these groups. For example, for the input array [4,3,6,8,2,3,1,5], the result should be [3,2,3,1,4,6,8,5] (because the average of these eight numbers is 4).

   You may use standard reduction and prefix operations without further detailed explanation. You should aim for $\Theta(\log\ n)$ run time, but describe your best attempt if you can't achieve this target.

   Comment on your algorithm's cost optimality (for example, if it is cost optimal, explain why, if it isn't discuss whether it can be made so by some kind of scaling).

   **Since the purpose of this exercise is to exploit prefix and/or reduction operations, please do not attempt to use any full sorting algorithm as part of your solution i.e. there must be no steps in your solution which say "we now sort [some part of the data]".**

   (60 *marks*)