ALGORITHMICS

Bachelor of Software Engineering - Course 2015-2016

SESSION 3.2

GOAL:

- **Divide and Conquer: parallelization**

PARALLELIZED QUICKSORT

The **Quicksort** algorithm is one of the best-known sorting algorithms. It allows to get **O(nlogn)** executions in the best and average cases.

In addition, algorithms designed using Divide and Conquer are very likely to be parallelized, i.e., to be executed using more than one processor of the machine at the same time, which can be highly recommended in today's computers, which easily have 4 or even 8 processors.

DEADLINE AND WORK TO BE DONE

You should implement, using as a basis the Divide and Conquer code you already have, a parallelized solution by using the **Fork/Join Framework**, available in Java from version 7. Name your new class **ParallelQuicksort.java**.

You are asked to design and implement an algorithm, as similar as possible to that provided, using the **central element as the pivot**.

To verify the advantages of the parallelized implementation, you are asked to include in the **QuicksortTimes.java** class the code you would need to measure times for the two versions of the code creating random vectors of size **n**

Study it empirically with measures of times for different sizes of the problem.

n	*Quicksort (t)*	*Parallel Quicskort (t)*
20.000	…..	…..
40.000	…..	…..
80.000	…..	…..
160.000	…..	…..
320.000	…..	…..
640.000	…..	…..
1.280.000	…..	…..
…..	…..	…..

You should deliver:

- *The source code files for* ***ParallelQuicksort.java*** *and* ***QuicksortTimes.java***
- *A document with the complexity, explaining its correspondence with the theoretical values and the table with times*

A task will be enabled in the virtual campus to upload the exercise. The deadline is 1 week and 1 day after the session ends.