Lab Session 04

Home exercises

1. **[5p]** Having **N** the number of elements, **P** the number of groups, **Tid** the identifier of the group (0, … , P-1), and **w** a power of 2, find a formula that gives you the following table for **N=32** and **P=4**. You can start from the formulas we used in the other labs:
   - **Start =** ceil**(N/P) * Tid**
   - **End =** min**(**ceil**(N/P) * (Tid**+1**), N)**
   - Hint: a solution is to approximate to a multiple of w.
   - Simplification: **N** and **P** are powers of 2.

	Tid=0		Tid=1		Tid=2		Tid=3	
	Start	End	Start	End	Start	End	Start	End
w=16	0	0	0	0	0	0	0	32
w=8	0	0	0	16	16	16	16	32
w=4	0	8	8	16	16	24	24	32
w=2	0	8	8	16	16	24	24	32
w=1	0	8	8	16	16	24	24	32

2. **[5p]** Write two sorting functions.
   - The first function sorts all the elements in row i of a matrix.
   - The second function sorts all the elements in a column j of a matrix.
   - You can make use of qsort.

Lab Exercises

1. **[10p]** Parallelize the **Merge Sort** algorithm.
   - **Hint:** Watch out for the interchange between v and vNew.
   - **Hint:** Do all frames need to perform merge in the last steps?
2. **[10p]** Perform **sanity check**, **stress test** and **scalability measurements**.
3. **[20p]** Starting from Bubble Sort build the parallel version of the **Odd-Even Transposition Sort** algorithm.
4. **[10p]** Perform **sanity check**, **stress test** and **scalability measurements**.
5. **[10p]** Build the **sequential** version of the **Row-Column Sort** algorithm.
6. **[20p] Parallelize** the **Row-Column Sort** algorithm.
7. **[10p]** Perform **sanity check**, **stress test** and **scalability measurements**.