

# Algorithm Engineering

22 June 2010

**Exercise 1 [10 points].** You are given  $K$  sequences of integers, having total length  $N$ , sorted in increasing order and stored on disk in  $K$  files. You have to design an I/O-efficient algorithm that finds the integers common to all  $K$  sequences, by assuming that the internal memory size  $M$  is able to contain  $(K+1) \cdot B$  integers (hence,  $K+1$  disk pages).

Evaluate also the I/O-complexity of the proposed solution.

**Exercise 2 [8 points].** Given a sequence  $S$ , let us define the *rank* of an item its *position* in the sorted  $S$ . Design a randomized algorithm that, given in input an integer  $k$ , finds in  $O(|S|)$  expected time the item of  $S$  having rank  $k$ .

**Exercise 3 [4+6+3 points].**

- Describe how it works the LZ-parsing of a string  $S$
- Explain how the suffix tree data structure can be used to efficiently compute the LZ-parsing of  $S$ .
- Illustrate the generic step of the algorithm of the previous item on the example: “ababc” has been already LZ-parsed and the suffix remained to be parsed is “abababd”. Execute **one-single step** via suffix tree.