# Algorithm Engineering

**11 February 2010**

**Exercise 1 [8 points].** Let Y a set of n keys of variable length, having distinguishing prefix D. Discuss the algorithms you can use to sort them and which are their time/space complexities.

**Exercise 2 [8 points].** Define the next(i) function of the Knuth-Morris-Pratt algorithm for a pattern P[1,p], and then compute its value on the pattern P[1,9]=abrarabra.

**Exercise 3 [8 points].** Let us given the set S= {5, 10, 3, 28, 29, 25, 17, 2}.

1. Design a storage for S based on Perfect hashing. Please indicate the universal hash functions used (with all their parameters), and the size of the tables.

2. Insert the keys of S in the solution you give in item (3.1).

**Exercise 4 [8 points].** Given the string T[1,9]=aaaabbccdd, compute the Canonical Huffman code for the letters of T assuming as probabilities their frequencies (semi-static model).

**[If you wrote notes on one of the above exercises, contact the teacher]**

Let H be a class of Universal Hash functions h: U -> {0,1,2,...,m-1}.

1. Let S be a subset of U formed by n keys. Prove that, if h is taken randomly from H, then the average number of collisions induced by h onto the keys of S is upper bounded by n^2/(2*m).

2. Use H to design a "fingerprinting scheme" that, taken a key of S, returns a fingerprint of O(log n) bits, and guarantees that the average number of fingerprints that are equal when applied on S is < 1.

**[OPTIONAL]** Design a <u>randomised</u> algorithm that COUNTS the number of repeated keys of X. The algorithm should be of Montecarlo-type (hence it may err) and take O(n) time and space [*hint:* use the fingerprint of item (2) above and the sorting of exercise (1).