IA167: String Algorithms	**Spring 2014**

Lecture 0:   Syllabus

Lecturer: Alex Popa   *Scribes: Alex Popa*

The topics marked with a "*" will be cover if there is enough time.

1. (a) Organization of the course: grading, syllabus.
    (b) Notation. Exact pattern matching. Naive algorithm.
    (c) Achieving O(n+m). Karp-Rabin algorithm.
    (d) O(n+m) in deterministic time. Knuth-Morris-Pratt algorithm.
    (e) * Boyer-Moore.

2. (a) Suffix trees.
    (b) Ukkonen's algorithm.

3,4 (a) Suffix arrays
     (b) LCP
     (c) RMQ in constant time

5,6 (a) Pattern matching with errors
     (b) Pattern matching with don't cares
     (c) Pattern matching with mismatches
     (d) Abrahamson algorithm for computing mismatches.
     (e) Wu-Manber algorithm.

7,8 (a) Edit distance vs longest common subsequence.
     (b) LCS in linear space.
     (c) Myers' algorithm for edit distance (algorithm used in 'diff' program from UNIX).
     (d) Computing LCS in subquadratic time (the algorithm of Masek and Paterson).

9 (a) Shortest Common Superstring
   (b) Approximation algorithms and the connection with the Travelling Salesman problem.

10 (a) Function matching
    (b) Parameterized matching
    (c) Generalized function matching

11,12 (a) Pattern matching in the streaming model where the text arrives online and our memory is less than the size of the pattern.
       (b) Porat&Porat algorithm solving the pattern matching in the streaming model using only O(log m log n) bits of extra memory.