- This problemset has *three* questions.
- To get the credit for questions marked as SPOJ, you must get them accepted on `http://www.spoj.com/AOS`, but you **don't** have to send any explanation!
- For other questions, either send the solutions to `gawry1+aos@gmail.com`, or leave them in the envelope attached to the doors of my office (room 321).

1. Given two strings s and t, we are interested in computing the length of their longest common substring, which is a string occurring in both s and t. Show how to solve this problem in linear time using the suffix array. You can assume that the lcp array is available, too.

2. Given a permutation on $\{1, 2, \ldots, n\}$, we want to find a word $w \in \Sigma^n$ such that its suffix array SA_w is exactly the given permutation.
   (a) Show a linear time algorithm solving the problem for $\Sigma = \{a, b\}$.
   (b) For extra credit: show a linear time algorithm solving the problem for $\Sigma = \{a, b, \ldots, z\}$.

(SPOJ) 3. Given two strings of length n, $p = p_1 \ldots p_n$ and $q = q_1 \ldots q_n$, define $M(i, j, k)$ as the number of mismatches between $p_i \ldots p_{i+k-1}$ and $q_j \ldots q_{j+k-1}$ (basically it is a substring Hamming distance). Given an integer K, find the maximum length L such that there exists pair of indices (i, j) for which we have $M(i, j, L) \leq K$.