

## HOMEWORK #4

---

**(5 points)** Implement the Burrows-Wheeler Transform (BWT) and its inverse. You may use any programming language and any BWT algorithm, but to receive full points:

- BWT *must run* in  $O(n \log^2 n)$  time in the *worst case* (it must be guaranteed subquadratic),
- and the inverse BWT *must run* in  $O(n)$  time in the *worst case*.

It is recommended to implement the Manber-Myers algorithm for suffix array construction. In the  $k$ -th phase of this algorithm, all suffixes are sorted according to the first  $2^k$  characters.

After obtaining a working implementation, verify correctness by testing that

$$bwt^{-1}(bwt(T)) = T.$$

Then, review your code and try to simplify or clean it up for clarity and efficiency.

Download the provided input files from:

Covid genome (30KB): <https://kubokovac.eu/ds/du/bwt/covid.txt>

The whole Shakespeare work (4.8MB): <https://kubokovac.eu/ds/du/bwt/SH.txt>

Human chromosome 21 (46MB): <https://kubokovac.eu/ds/du/bwt/chr21.txt>

(this is the shortest one)

For each input file, compute and report:

- Original length of the text.
- Number of runs in  $bwt(T)$  – a *run* is a maximal substring consisting of a single repeated symbol (for example AAACBBB contains 4 runs).
- Execution time for:
  - computing the BWT, and
  - computing the inverse BWT.