Homework #4

(3 points) Implement BWT transform and inverse BWT transform. You can use any programming language and implement any BWT algorithm, however, for full points, BWT must run in $O(n \log^2 n)$ time in the worst case (we want less than quadratic time guaranteed) and inverse BWT must run in $O(n)$ time in the worst case.

It is advised to implement the Manber-Myers algorithm for suffix array construction where in the k-th phase, we sort all suffixes according to the first 2^k letters.

Test your program: make sure that $bwt^{-1}(bwt(T)) = T$. After achieving a working implementation, go through your program once again and try to clean it up / simplify it.

Download the input files for testing from

`https://people.ksp.sk/~kuko/ds/du/bwt/`

and for each input calculate:

- the original length,
- number of runs in $bwt(T)$ (a *run* is a substring consisting of a single repeated symbol),
- measure the time to compute BWT and the time to compute inverse BWT.