

Let $y = y[0..n-1]$ be a string on a fixed-size alphabet.

Design, implement and analyse efficient algorithms for:

- (1) computing **all local periods** of y (the local period at position i on y is the smallest length of a nonempty word w satisfying: one of w and $y[0..i-1]$ is a suffix of the other, and one of w and $y[i..n-1]$ is a prefix of the other);
- (2) counting and displaying **all runs, squares and cubes** occurring in y (a run in y is a factor $y[i..j]$ whose length is at least twice its smallest period, and for which $y[i-1..j]$ and $y[i..j+1]$, when they exist, have larger periods; squares and cubes are factors of the form vv and vvv respectively);
- (3) displaying **all critical positions** of runs in y (a position i on u is critical if the local period at i is the smallest period of u).