

Homework 5 (Rcpp part I)

Select and solve **5** tasks from this list.

Include all the solutions in **one** R script file, see `homework5_template.R` for a template.

When you're done, send them via courses.ipipan.edu.pl.

All the scripts will be examined by plagiarism detection software.

Exercise 05.01. Write an Rcpp function `lcs()` to compute the length of the longest common subsequence of two numeric vectors.

Exercise 05.02. Write an Rcpp function `sortedmerge()` which merges two already sorted (nondecreasingly or nonincreasingly – two cases are possible) numeric vectors into a one, sorted vector. If incorrect data is provided, call `stop()` to throw an error.

Exercise 05.03. Write an Rcpp function `naomit()` to remove all missing values from a given numeric vector.

Exercise 05.04. Write an Rcpp function `sample2()` to generate a random subvector of length `k` of a given numeric vector `x` (without replacement, something like `sample(x, k)`).

Exercise 05.05. Write an Rcpp function `randperm()` to generate a random permutation of a given numeric vector.

Exercise 05.06. Write an Rcpp function `NAimput()` to impute all missing values found in a given nondecreasingly ordered numeric vector `x`.

If there are any missing values at the beginning or at the end of `x`, substitute them, respectively, for the first or last non-missing value. For example, if the input is `(NA, NA, 4, 5, NA)`, then the output should be `(4, 4, 4, 5, 5)`

Otherwise, use a linear interpolation between the neighboring elements. For example, given `(1, NA, 2, NA, NA, 3)`, we expect to get `(1, 1.5, 2, 2.33, 2.67, 3)`.