

Exercise Sheet 3

Submit until Tuesday, November 10 at **2:00pm**

For this exercise sheet, you can use Java or C++, but not Python. Subtle algorithmic improvements and performance tuning make little sense in Python, due to the large overhead of the underlying data structures, in particular the Python lists/arrays.

Copy the code from the lecture and proceed from there. It is available in both Java and C++. It contains code for reading posting lists from the given files, for the baseline intersection, and for time measurement.

Exercise 1 (10 points)

Implement a method for intersecting two posting lists that uses at least three non-trivial ideas presented in the lecture. The goal is to beat the baseline implementation from the lecture for all scenarios, in particular for two lists of roughly equal size as well as when one list is much shorter than the other. Note that you can also implement several algorithms and switch between them depending on the sizes of the input lists (or depending on any information that you find to be useful).

Exercise 2 (10 points)

Evaluate your algorithm on all three combinations of the three posting lists on the Wiki. Report the results from your evaluation in the table on the Wiki, following the examples of the rows already there.

As usual, summarize your experiences and insights in your *experiences.txt*. Try to be brief but informative. Test, Checkstyle, Jenkins all as usual.

Exercise 3 (optional)

Prove that there are $n + k$ choose k ways to “place” k elements within a list of n elements.

Add your code to a new sub-directory *sheet-03* of your folder in the course SVN, and commit it. As usual, also commit a text file *experiences.txt* where you briefly describe your experiences with this exercise sheet and the corresponding lecture. As a minimum, say how much time you invested and if you had major problems, and if yes, where.