Advanced Data Management for Data Analysis

Assignment 3 *(individually!)*

02.11.2022

Due: ***Tuesday, 15 November 2022, 09:00 CET***

Notes:

- Write (preferably) **C** or **C++** programs as detailed below.
- Write a PDF document (called "**report.pdf**"; preferably max. two A4 pages) discussing your programs as detailed below.
- Create a compressed archive containing your **report.pdf**, your programs' entire source code as well as any accompanying scripts (if any).
- Name your submission file "**ADM2022_A3_<student_id>.zip**"
- Submit via BrightSpace

Points:

This assignment is worth a total of 100 points, 50 points for the programs and 50 points for the report discussing the programs (see details below). The final score (grade) will be points divided by 10 to fit in the 0-10 grade system.

Please read the entire assignment instructions carefully and make sure you follow them properly.

Your task is to write a (preferably) **C** or **C++** program that reads a sequence of $n \in [1,2147483647]$ non-negative 4-byte integer numbers (in the range $[0,2^{31}-1]$, i.e., [0,2147483647]) from a given file into an in-memory array, **sorts** that array in memory, and then outputs the sorted array to the console (stdout). You can use any **comparison-based** sort algorithm you prefer; the absolute sorting performance is not the main concern, here.

Implement **two versions** of your program, **one that (potentially might) suffer(s) from branch mispreditions** (as discussed during the course's lecture on Wednesday November 02, 2022), and **one that surely does not suffer from branch mispredictions**.

In your report (in PDF; called **report.pdf**), explain the general approach of your algorithm(s) as well as why the first one might suffer from branch mispreditcion and how you avoid them in your second implementation. Discuss the complexity of both algorithms. Measure and report the time each of your programs/algorithms takes to sort the array (*excluding* the time it takes to load the data from file into memory and to output the result to the console (stdout)). Please also briefly explain how your programs need to be compiled, and how to run your programs.

Your programs must accept two command line arguments:

1. the number of values to read from the file, i.e., the size of the in-memory array (a non-negative 4-byte integer in the range $[0,2^{31}-1]$, i.e., [0,2147483647]),
2. the (absolute or relative) path to the file containing the data.

For your convenience, program templates in C that implement everything but the actual sorting algorithm are provided at
https://homepages.cwi.nl/~manegold/ADM/ADM-2022-Assignment-3-C-code-templates.zip

As sample data, please feel free to use any or all of the l_*int*.csv files provided with Assignment 2 (yes, all their actual values fit in range $[0,2^{31}-1]$, i.e., [0,2147483647]).

Pack the source code of your programs and your report in a single archive for submission:
ADM2022_A3_<student_id>.zip