External sorting algorithm

Project for the "Data Structures and Algorithms – Practicum" Course

This is a C++ implementation of the External sorting algorithm used for sorting huge amounts of data that is unable to fit into RAM at one time.

This algorithm basically consists of two main phases:

1. ***Partition:*** The data is partitioned into smaller "chunks", each of which is small enough to fit into RAM. We read as many numbers from the input file as our memory can handle at a time. We then sort these numbers and write them in a file.
2. ***Merge:*** Using the K-way merge algorithm, the smaller "chunk" files (already sorted) are merged into one, which becomes our output file.

How the program works:

1. User writes the names of the input and output file and the memory capacity they want the program to use into the console.
    - There is a generator in the project that can be used for creating input files with given name and number of integers.
2. When the process of sorting the given data and storing it into the output file finishes, the program prints a message containing the execution time it took.

How was the algorithm tested?

1. File "input_20k.txt", containing 20,000 integers, with total file size 234 KB was sorted for ca. 0.50 seconds using chunks of 100 MB RAM.
2. File "input_500k.txt", containing 500,000 integers, with total file size 5.71 MB was sorted for ca. 11 seconds using chunks of 100 MB RAM.
3. File "input_2mil.txt", containing 2,000,000 integers, with total file size 22.8 MB was sorted for ca. 45 seconds using chunks of 100 MB RAM.
4. File "input_5mil.txt", containing 5,000,000 integers, with total file size 40 MB was sorted for ca. a minute and a half using chunks of 100 MB RAM.
5. File "input_10mil.txt", containing 10,000,000 integers, with total file size 114 MB was sorted for ca. three and a half minutes using chunks of 100 MB RAM.
6. File "input_150mil.txt", containing 150,000,000 integers, with total file size 1.67 GB was sorted for ca. 55 minutes using chunks of 100 MB RAM.
7. File "input_580mil.txt", containing ca. 580,000,000 integers, with total file size 4.52 GB was sorted for ca. two and a half hours using chunks of 100 MB RAM.

Author: Georgi Sabev, 62380