Programming Assignment 5

CMPE 250, Data Structures and Algorithms, Fall 2014

Instructor: A. T. Cemgil
TA's: Barış Kurt, Atakan Arıkan

Due: 10 January 2015, 23:59

External Sorting

Sorting is the most fundamental algorithmic problem in computer science and it's usually the first step of the solutions of many large scale problems. Therefore, many sorting algorithms have been invented for different purposes. With the recent developments in networking and mobile services, we are constantly generating and processing huge amounts of data that is impossible to fit into a single memory. Instead, we use external sorting methods.

External sorting is a term for a class of sorting algorithms that can handle massive amounts of data. External sorting is required when the data being sorted do not fit into the main memory of a computing device (usually RAM) and instead they must reside in the slower external memory (usually a hard drive).

External sorting typically uses a hybrid sort-merge strategy. In the sorting phase, chunks of data small enough to fit in main memory are loaded to memory, sorted, and written out to a temporary file. In the merge phase, the sorted data portions in the temporary files are merged into a single larger file.

1. Read data from the disk that can fit into your memory.
2. Sort the data with a conventional sorting algorithm.
3. Write the sorted data to a temporary file on the disk.
4. Repeat steps [1-3] until all the data is partially sorted.
5. Merge the data in the temporary files into a single file.

Assignment

You are going to be given a text file containing a few thousands of double precision numbers and you will not be allowed to use memory of size $8 \times 1024 = 8K$ bytes, which means you can load at most 1024 double precision numbers into your memory at once.

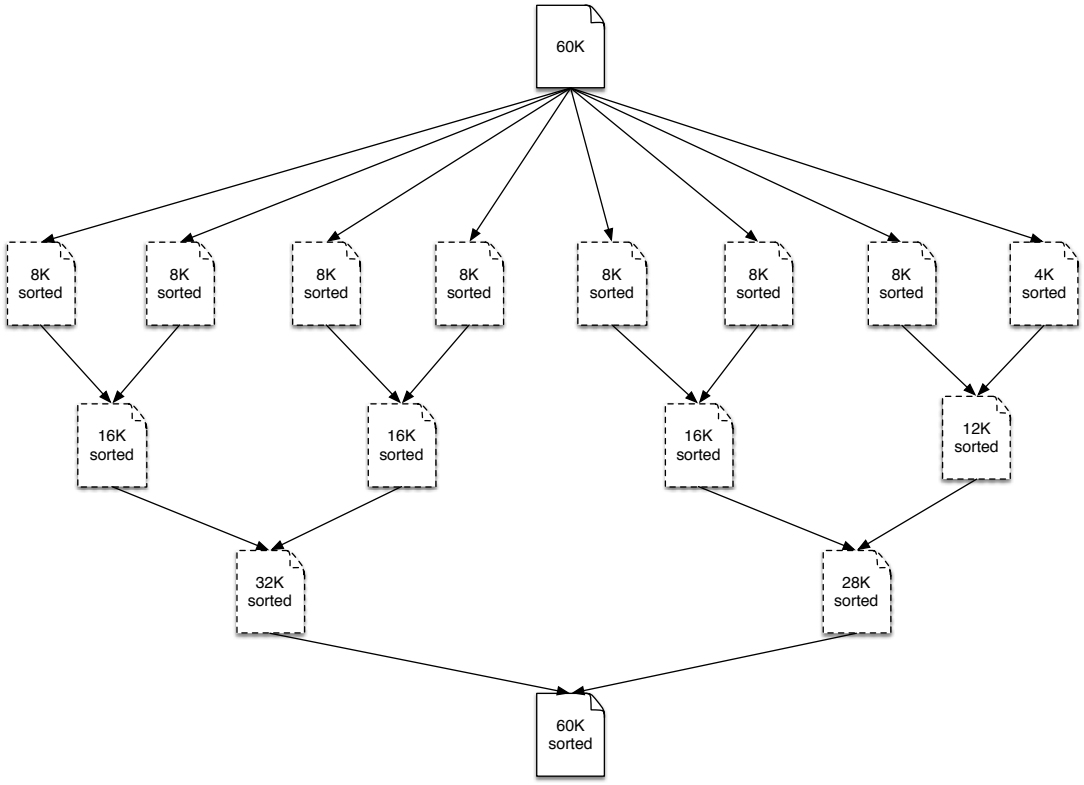

Figure 1: 2-way external sorting of a file of size 60K bytes. The temporary files are drawn dashed.

Rules:

- You need to implement your own sorting algorithm for step 2. It's recommended that you implement one of the $O(n \log n)$ algorithms like HeapSort, MergeSort, or QuickSort. But remember the memory rule: you cannot exceed your memory limit. For example, if you use MergeSort, you can sort at most 512 numbers, since the sort algorithm itself uses $O(n)$ space for merging. You can use recursive QuickSort although it uses $O(\log n)$ memory in the stack space.
- You are free to implement any merging algorithm. Figure 1 shows an example of 2-way external merge sort. You can use different merging methods if you want.
- You cannot use any containers in the STL or any 3rd party library. This means using vector, map, set, queue, priority queue, etc... is forbidden. If you want to implement HeapSort, you can use *std::make_heap()*, *std::push_heap()*, *std::pop_heap()* functions from the *algorithm* library.
- If you do not make external sorting, or do not write your own sorting function you are going to get negative points for cheating.

Input/Output

Your algorithm will be tested with the following command:

```
./project5 [input_file] [output_file]
```

The input file contains the unsorted double precision numbers, one at a line. The first line in the input file will be the total number of numbers. The output file will contain those numbers in increasingly sorted order.

Temporary Files

There's no limit on the number of temporary files you can open. But you can create temporary files under the /tmp directory only. In Linux systems, /tmp is specially designed to store temporary files and it's cleared at reboot. However, in this project you are supposed to delete your own temporary files immediately when they are no more required. In C++, there is a *remove()* function for deleting files which is defined in *cstdio* library.

Report

This time, you are asked to write a 1 page report in *.pdf* format to explain your strategy in solving this problem. You need to be specific and clear about which sorting algorithm you used to sort small parts, and how did you merge the temporary results. What is the total runtime of your design? How much extra external memory do you use? You shoud submit this document also via git.

Submission Details

You are supposed to use the Git system provided to you for all projects. No other type of submission will be accepted. Also pay attention to the following points:

- All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code.
- You are expected to use C++ as powerful, steady and flexible as possible. Use mechanisms that affects these issues positively.
- Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.