

# Lab 4 in C++ OOP

Hergeir Winther Lognberg  
Hewi1600

## 1 Preamble

The Lab contains folders, each containing an header and corresponding cpp file. The names of the folders correspond to the classes they contain. I'm very sorry for not keeping the main clean, but simply haven't gotten the time.

### 1.1 Compilation

For compiling i use g++ and created a makefile to make the compiling process easier. Just run *"make"* within the lab directory compile.

## 2 The Software

I've tried to keep redundancy away from my code. Each class is only written once and reused in this lab.

Also I have tried to use reference and *const* everywhere it made sense to do so.

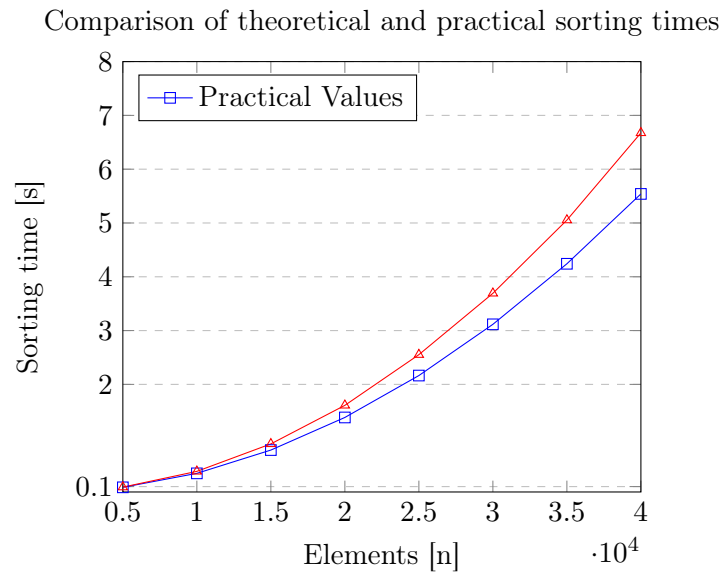
I've tried to use what i've learned so far in an effecient way. I've used an enum called *Algorithm* to keep track of sorting algorithms. Also I've used a struct to bind data together (averageSortingTime,arrayElements and algorithm). Also I created my own Array class from scratch before I read the assignment thoroughly. It dynamically increases and decreases array by deleting and reallocating. I don't have any comment regarding the sorting functions as they where not my creation. I chose to put all timers inside the sorting functions themselves to get rid of function calling-time.

Would love any feedback on good code practice.

## 2.1 Results

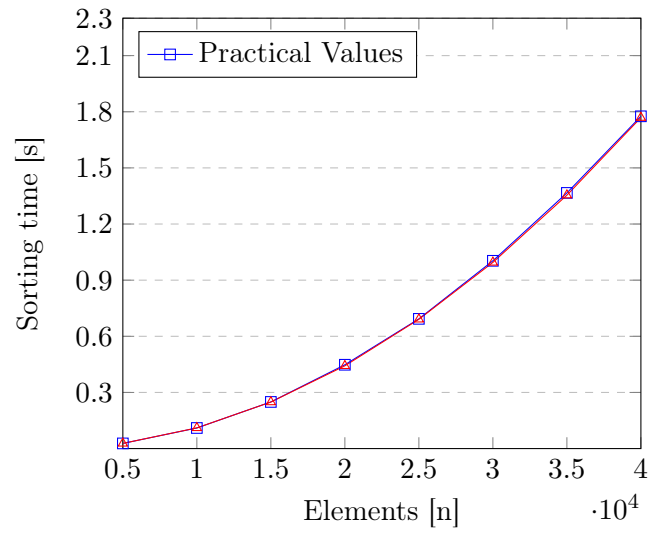
In I've made graphs for all sorting algorithms and included the theoretical graph for each of them. All in all I see that theory is indeed very close to reality i my case. The only algorithms to visibly deviate from theory are BubbleSort and QuickSort (the two extremes). BubbleSort seemed to have some overhead unaccounted for, while quickSort was quicker than theory would allow.

### 2.1.1 BubbleSort



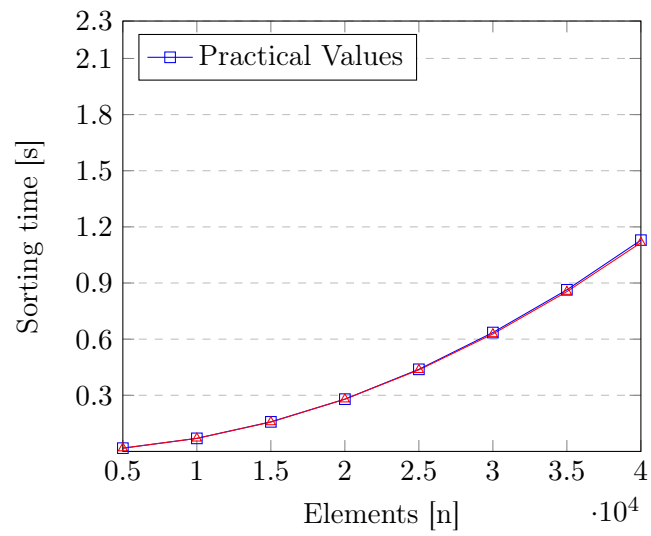
### 2.1.2 SelectionSort

Comparison of theoretical and practical sorting times



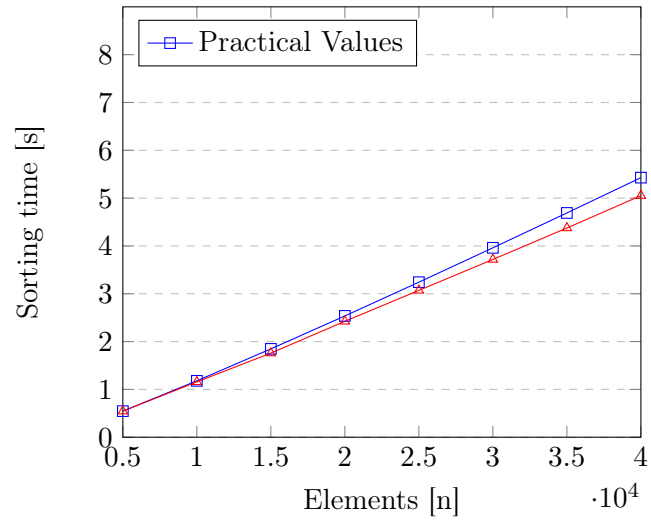
### 2.1.3 InsertionSort

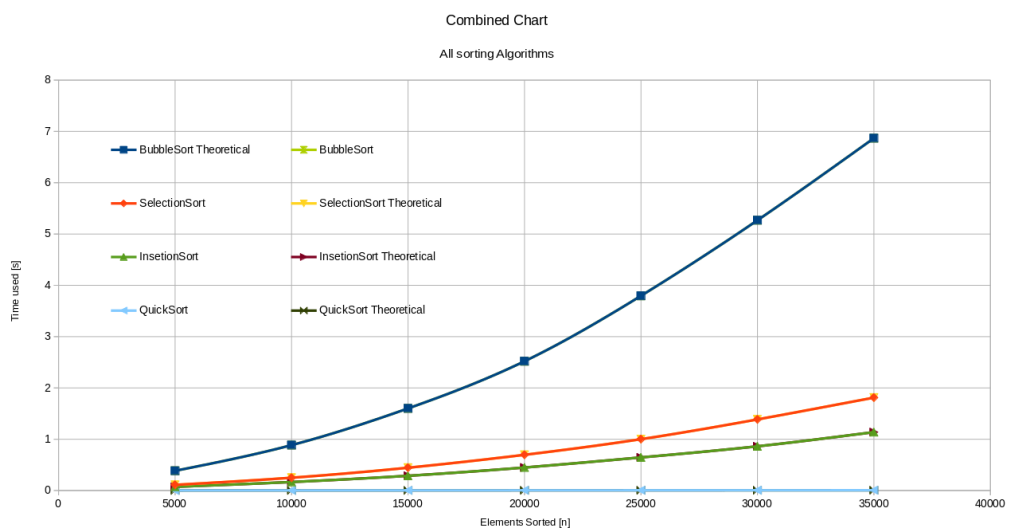
Comparison of theoretical and practical sorting times



#### 2.1.4 QuickSort

Comparison of theoretical and practical sorting times  
·10<sup>-3</sup>





Here we see that in everyway imaginable. That amongst the sorting algorithms we tested, Quicksort rules them all.

### 3 Enviroment

I'm programming on an Arch Linux 64-bit system. I've got the c++ compiler installed and compile using it's g++ alias which links necessary libraries automatically. To compile i use the recommended flags: "-std=c++11 -Wall -pedantic". The flags let me choose to use c++11 standard and give me useful compiling warnings and errors.

Monday 13<sup>th</sup> February, 2017