Numerisches Praktikum – Numerical Practical Training

Hubert Klahr, Bertram Bitsch

Sorting

Return by 9:15 a.m. on Monday

Free Training

- Write routines using the

  1. straight insertion
  2. Shell's
  3. quicksort

  algorithms presented in the lecture[1]. Use the following array to test your routines

$$[7, 5, 3, 1, 9, 6, 10, 2, 8, 4]\,.$$

Assignment for the Afternoon / Homework

- **Exercise 1, 5 points**: Verification
  Verify that your algorithms work using the above list and another list of 10 random number. Print out the lists before and after sorting.

- **Exercise 2, 10 points**: Timing on *unsorted* lists
  Measure the runtime[2] of your algorithms for unsorted lists[3] of the length $N = 10^n$ with $n = [2, 3, ..., 8]$, if feasible. Discuss the occurred and possible problems. Plot the results in a double-logarithmic diagram. What are the scaling properties?

- **Exercise 3, 5 points**: Timing on *sorted* lists
  Do the same (Ex. 2) for perfectly sorted lists (i.e. $A = [1, 2, 3, ..., N]$).

[1] To compare your program, you can get a copy of the non-recursive quicksort routine for C/C++ at `https://code.woboq.org/userspace/glibc/stdlib/qsort.c.html`.

[2] Use internal timer functions like `clock()` for C/C++.

[3] Use the built-in function `rand` in C/C++ and FORTRAN (check `RAND_MAX` for the largest number generated).