Sorting Algorithms in C

This project contains various sorting algorithms implemented in C. Each are tested in terms clocks and with an ascending, descending, and random permutation of integers from `1` to `n`.

File Structure

This project contains two subdirectories: `algos` and `lib`.

Algorithms

The `algos` directory contains the C source codes for each sorting algorithm:

- `bubble_sort.c`
- `insertion_sort.c`
- `selection_sort.c`
- `shell_sort.c`
- `heap_sort.c`
- `merge_sort.c`
- `own_quick_sort.c`
- `hoare_quick_sort.c`
- `built_in_quick_sort.c`

The `gen-algos` directory contains the C source codes for each sorting algorithm but for generic data type:

- `gen_bubble_sort.c`
- `gen_insertion_sort.c`
- `gen_selection_sort.c`
- `gen_shell_sort.c`
- `gen_heap_sort.c`
- `gen_merge_sort.c`
- `gen_own_quick_sort.c`

Library

The `lib` directory contains header files and function implementation of common routines used to implement and test the sorting algorithms:

- `template.c`: Template file can be copied in the `algos` directory to implement a new sorting algorithm.
- `utils.h`: Header file that contains common utility functions for creating random permutations and implementing sorting algorithms.
- `utils.c`: C file that implements the functions in `utils.h`.
- `brute.h`: Header file that contains functions for generating all permutations of `1` to `n` in lexicographical order and brute force testing a sorting algorithm.
- `brute.c`: C file that implements the functions in `brute.h`.

Scripts

The project contains bash scripts to automatically compile and execute the sorting algorithms.

- `run.sh`: Script to compile and run the given sorting algorithm C program.
- `run3x.sh`: Script to compile and run the given sorting algorithm C program thrice.
- `runall.sh`: Script to compile and run all sorting algorithm C programs.
- `runall3x.sh`: Script to compile and run all sorting algorithm C programs thrice.

How to Use

Compilation and Execution:

To compile and run the program, use the scripts as follows:

```
./run.sh <filename> <arraysize> asc # ascending permutation
./run.sh <filename> <arraysize> desc # descending permutation
./run.sh <filename> <arraysize> rand <seed> # random permutation
./run3x.sh <filename> <arraysize> asc # ascending permutation 3x
./run3x.sh <filename> <arraysize> desc # descending permutation 3x
./run3x.sh <filename> <arraysize> rand <seed> # random permutation 3x
./runall.sh <arraysize> asc # ascending permutation
./runall.sh <arraysize> desc # descending permutation
./runall.sh <arraysize> rand <seed> # random permutation
./runall3x.sh <arraysize> asc # ascending permutation 3x
./runall3x.sh <arraysize> desc # descending permutation 3x
./runall3x.sh <arraysize> rand <seed> # random permutation 3x
```

Make sure to give executable permission using:

```
chmod +x <scriptfile>
```

Example:

To run insertion sort in the C program `insertion_sort.c` with a random array using `seed = 7` and an array size of `n = 10000`, use the scripts as follows:

```
./run.sh insertion_sort 10000 rand 7
```