Program

Usage

Arguments

- opt = optional
- sorting_methods (default merge): `merge`, `bubble`
- files in this repo is: `1.txt`, `2.txt`, `3.txt`, `4.txt`
- `loud` tells the program to output the sorted array

```
$ ./program [file_to_search] [opt: sorting_method] [opt: loud]
```

Run example

Run program

run `make clean && make` to recompile

Standard merge sort:

```
$ ./program data/4.txt
>(sorting with merge took 45 ms.)
>Number to search for (0 to skip): 998869
>index of 998869 in the sorted array is 199766 (used to be index 114504)
```

Specify other sorting methods:

```
$ ./program data/3.txt bubble
```

Print sorted array to output with `loud` (requires specifying sorting as well)

```
$ ./program data/4.txt merge loud
```

Run tests

```
$ make test_cycle
```

Implementation

Sorting

I have included Bubble-sort and Merge-sort. They are both pretty standard, and it's not much to say

about them. I chose Merge-sort because it's a fast sorting method with a stable *O(n*log(n))* performance, and i chose it over Quicksort simply because I prefer a better worst-case runtime over low memory usage, and I was more familiar with Merge-sort.

Searching

Searching has a standard Binary Search. This is the the fastest searching-algorithm on a sorted list by far, with a worst case *O(log(n))* performance. There were no need to implement a quicker searching-method

FileElement

FileElement is a object read from a file, and it simply contains a value along with its original index in the file.

DynArray

DynArray(Dynamic Array) is a wannabe *Java.util.ArrayList*, made for *FileElements*. It doubles the size once it's full, up to its capacity, and allows memoryhandling-free interaction with the array. It's pretty straight forward.

Helpers

All helper-methods has been put here. They mainly assist *Program.c*.

Questions

1) Is it OK to include the header files and their respective .c files in the "main file"? Or should they be included where they are used? 2) Should the helper-methods (*/helpers/helpers.c*) be implemented in its own class like I did, or should they just have stayed in *Program.c* ?

Sources

Sources are also commented where they are used.

- Getting timestamp in milliseconds: http://stackoverflow.com/questions/3756323/getting-the-current-time-in-milliseconds
- Dynamic array implementation: http://www.happybearsoftware.com/implementing-a-dynamic-array.html
- Merge-sort is inspired by the sudo-code here: http://www.sorting-algorithms.com/merge-sort
- Makefile inspired by: http://c.learncodethehardway.org/book/ex2.html