Algorithm and Data Structure

TD2 : Implementing searching and sorting algorithms

Objectives

Implementing searching and sorting algorithms you saw in the course.

Master the usage of `malloc` and `free`.

---

1 Searching Algorithms

1. Implement the two functions `is_sorted_nondecreasing` and `min_int` in `utils.c`
2. Fill the implementation of the three searching algorithms : linear search, jump and binary search. For binary search the implementation should be iterative not recursive. For both jump and binary search, the function should check if the array is sorted, if not return -1.
3. Compare the runtime of each algorithm in **seconds**.

2 Sorting Algorithms

1. Implement the three functions `swap_int`, `is_sorted_nondecreasing` and `copy_array` in `utils.c`
2. Fill the implementation in an iterative way for the three simple sorting algorithms : selection, insertion and bubble sorts.
3. Fill the implementation in a recursive way for the two complex sorting algorithms : merge and quick sorts.
4. Compare the runtime of each algorithm in **seconds**.