

Implementing heapsort versus selection sort as a single algorithm - with two different data structures

May 10, 2023

1 Project description

You are going to implement a program that implements two sorting algorithms: selection sort and heapsort. But you are **not** going to implement two different sorting algorithms. Instead, you are going to implement a single algorithm **with two different data structures**:

- A vector. In this implementation the operation FindMaxAndSwap (finding the maximum element and swapping it on the last position) takes $\Theta(n)$ time.
- A heap. In this implementation the operation FindMaxAndSwap (finding the maximum element and swapping it on the last position) takes $\Theta(1)$ time. Of course, to maintain the heap property you have to call Heapify, which takes a further $\Theta(\log n)$ time.

2 Guidelines

Your program should have the following basic functionality:

- Create two different data structures corresponding to the two different implementations. If you use an object-oriented language: Each of them should have the needed member functions. Data should be private, and accessible from the main program only through calling member functions. You have freedom to the design of the two classes. **Discuss your design choices in the report you submit.**
- The main program will have two identical functions Sort. They will differ through the type of the parameter, corresponding to the two different classes (Heap/Vector)
- The main function should call these two functions with two different objects **corresponding to the same sequence.**

- You should time the two sorting method on random permutations of a sufficiently large size. Python provides the means to generate such large random permutation.

3 What to turn in

You will turn in

1. *a short written report* containing:
 - A description of the significant choices/issues in the design of your code.
 - Listing of your experiments, showcasing the capabilities of the program.
2. The sourcecode of your program.

You may turn in the document via moodle.unibuc.ro (Class moodle).

4 Coding standards

A percentage of your grade will be based on the quality of your code, so pay attention to it. Discuss changes (if any) you made to programs presented in class. Take extra care in documenting the code you are implementing on your own. Properly modularize the code (for instance implement separate functions for significant parts of the program).

5 Deadline

June 2, noon local time. This is a strict deadline. No credit will be given for homework turned in late.