DSSC - Algorithmic Design

Homework 3

Patrick Indri

May 14, 2019

1 Introduction

The aim of this assignment is to implement an array-based representation of a binary heap. An iterative version of the `Heapify` version should be implemented as well. The `code/` folder contains such implementation.

Compile and run

The `makefile` provided can be used to compile the code. Using `make` or `make all` the code gets compiled in a verbose version `main.x`, useful to test the implemented functions and see their output. Running `make benchmark` results in the compilation of `benchmark.x` which benchmarks the code and provides a cleaner output, ready to be plotted.

2 Implementation

The code implements all the basic functions to deal with heaps and revolves around the `Heapify` function, implemented as a recursive function. Functions get the root element, remove it and verify the heap property have been implemented as well. It should be noted that, as presented, the code implements a min-heap. However, editing the `HEAP_ORDERING` macro in `heap.c` allows the implementation of a max-heap. The `BuildHeap` function takes a randomly ordered array and heapifies it. Its performance will be evaluated in the following section.

3 Benchmark

Running `benchmark.x` tests the performance (and the correct behaviour) of the `BuildHeap` and `RemoveMinimum` functions, timing their execution.

Fig. 3.1

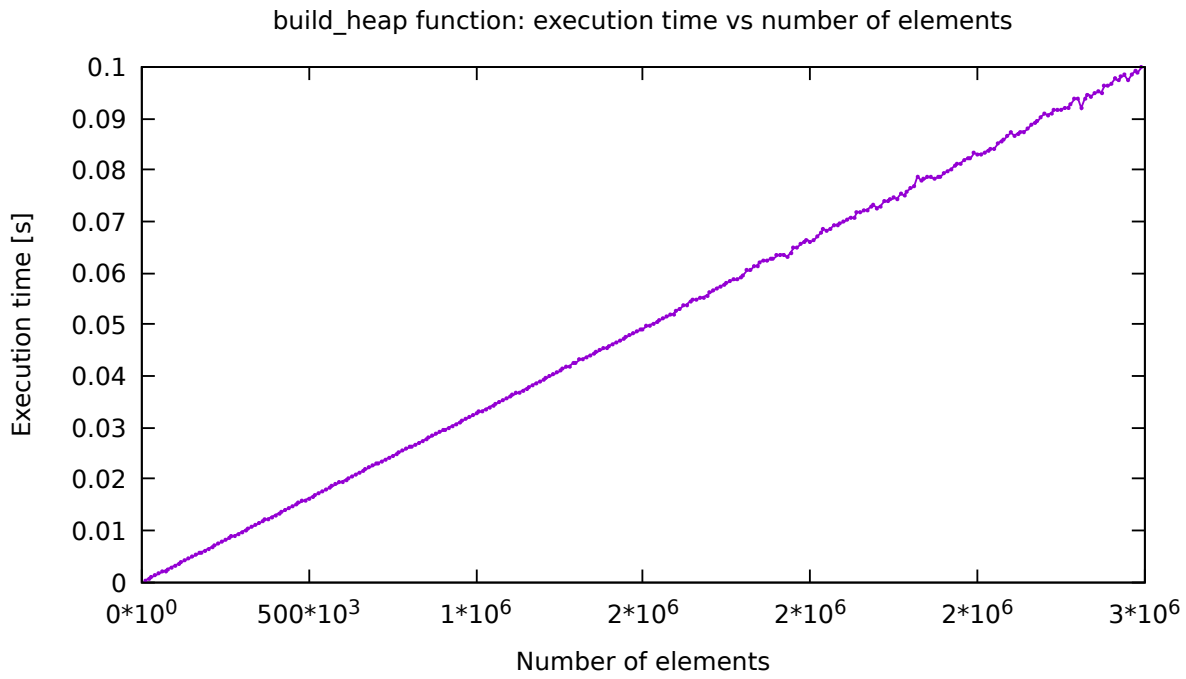

Fig. 3.1 shows the results of the benchmark. As expected, the computational cost of the execution of the `BuildHeap` function is $O(n)$, where n is the number of nodes of the heap.