Exercises — Heap sort

version #

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.assistants.epita.fr

1 Heap sort

Files to submit:

- heap_sort/heap_sort.c
- heap_sort/heap_sort.h

Authorized headers: You are only allowed to use the functions defined in the following headers:

- err.h
- assert.h
- errno.h
- stddef.h

1.1 Goal

In this exercise you will have to implement two algorithms about the heap data structure: `heapify` and `heap_sort`.

The main idea behind `heapify` is to construct a max-heap. A max-heap is a complete binary tree whose biggest value is at the root and in which each children is smaller than its parent.

`heap_sort` is an algorithm to sort a list by using `heapify`. You need to swap the root of the heap with the last element of it, placing the biggest element of the array at the end. Then, you need to transform again the `n - 1` first elements of the heap repeat the process on them until you get the list sorted.

1.2 Heapify

Write the following function:

```
void heapify(int *array, size_t size);
```

This function transforms in place an integer array to a max-heap. It is useful when you need to update a list after swapping elements. For this exercise, you can assume that `array` will never be `NULL`.

1.3 Sort

Write the following function:

```
void heap_sort(int *array, size_t size);
```

This function sorts the elements of the input integer array in place by ascending order using the `heap sort` algorithm. For this exercise, you can assume that `array` will never be `NULL`.

It is my job to make sure you do yours.