

# Heaps - Recitation 4

Correctness, Loop Invariants And Heaps.

November 5, 2022

## Abstract

Apart of quantify the resource requirement of our algorithms we are also interested to prove that they indeed work. In this Recitation we will demonstrate how to prove correctness via the notation of loop invariant. In addition we will present the first (non-trivial) data structure in course, and prove that it allows us to compute the maximum efficiently.

## 1 Correctness And Loop Invariant.

In this course, any algorithm is defined relative to a task/problem/function, And it will be said correct if for any input it computes desirable output. For example, suppose that our task is to extract the maximum element from a given array. So the input space are all the arrays of numbers, and proving that a given algorithm is correct, requires from us to prove that for an arbitrary array the algorithm's output is the maximal number. Formally:

**Correctness.** We will say that an algorithm  $\mathcal{A}$  (an ordered set of operations) computes  $f : D_1 \rightarrow D_2$  if for every  $x \in D_1$  the following equality holds  $f(x) = \mathcal{A}(x)$ . Sometimes when it's obvious what is the goal function  $f$ , we will abbreviate and say that  $\mathcal{A}$  is correct.

Examples of functions  $f$  might be: file saving, summing numbers, or posting a message in the forum.

Usually it will be convenient to divide the algorithms into subsections and then characteristic, and prove correctness for each of them separately. One main technique is using the notation of Loop Invariants.

**Loop Invariant.** Loop Invariant is a property that characterizes a loop segment code and satisfies the following conditions:

1. Initialization. The property holds (even) before the first iteration of the loop.
2. Conservation. As long as one performs the loop iterations, the property still holds.
3. (optional) Termination. Exiting from the loop carrying information.

**Example.** Before dealing with the hard problem, let us face the naive algorithm to find the maximum of a given array.