

Introduction to Algorithms

Reading

- Read 1.1, pages 3-7
 - What is an **algorithm**? Be very clear about the key points that would constitute an algorithm. Give examples of solutions to a problem that would NOT be considered algorithms.
 - What is an essential requirement for every step of an algorithm?
 - Is it possible for a problem to have many algorithms for its solution?
 - List the steps of Euclid's algorithm for computing the greatest common divisor of two numbers. Also write the algorithm in pseudocode.
 - * What happens to Euclid's algorithm when the first number is smaller than the second?
 - Two key ways of describing an algorithm is via a sequence of steps and via pseudocode. Describe each way and the advantages and disadvantages.
 - Describe the *sieve of Eratosthenes* algorithm for finding a list of prime numbers.
 - Challenges (optional): Exercises 1.1.11, 1.1.12.
- Read 1.2, pages 9-16
 - Correct the order of the following steps that are needed when trying to find an algorithmic solution to a problem, and add any other needed steps:
 - * Understand the computational means of our system
 - * Choose a strategy for the algorithm
 - * Clearly define what the inputs and outputs in an instance of the problem are
 - * Try some simple cases
 - * Write code for the algorithm
 - * Prove that the algorithm is correct
 - * Specify the algorithm via pseudocode or a flowchart
 - * Analyze various aspects of the algorithm like efficiency, simplicity, generality
 - What are the two kinds of efficiencies that we look at regarding an algorithm?
 - What are the differences between sequential and parallel algorithms?
- Read 1.3, pages 19-23
 - What do we refer to as a **sorting problem**?

- Give at least 3 examples of sorting from real life (i.e. situations where values are presented in an increasing or decreasing order based on some key, and where this ordering presents some advantage).
- When is a sorting algorithm called **stable**? Be precise.
 - * What is the advantage of stable sorting algorithms
- When is a sorting algorithm called **in-place**?
- What are other important classes of problems?
- Work on exercise 1.3.8
- Challenge: 1.3.9