

Introduction

(CLRS 1)

- Class is about *designing* and *analyzing algorithms*
 - *Algorithm*: A well-defined procedure that takes an input and computes some output (not code, does not contain the low-level details).
 - *Design of algorithms*: Methods/ideas for developing (efficient) algorithms. There are no recipes, but there exist some general techniques and we'll study them.
 - *Analysis of algorithms*: Abstract/mathematical comparison of algorithms (without actually implementing them). Think of analysis as a measure of the quality of your algorithm and use it to justify design decisions when you write programs.
- In this class we do all these:
 - come up with solutions for a problem
 - prove that it is correct
 - analyze its running time
- Hopefully the class will show that **algorithms matter!**

Warmup: searching and sorting

The fundamental problems are searching and sorting.

- Searching: Given an input array A consisting of n elements, and a target element say t , find if t occurs in A (yes/no answer).
 - for simplicity we assume integers
 - variations: find index where it occurs; find all occurrences; count number of occurrences.
- Sorting: Given an input array A consisting of n elements, permute the elements such that the sequence is now sorted, i.e. $A[0] \leq A[1] \leq A[2] \leq \dots$

We'll warm up with the following algorithms:

- searching: linear search, binary search
- sorting: bubble sort, selection sort, insertion sort