

DSAT (Lecture 1)

Administrative Details:

Grading:

- 30% : Two assignments each 15% (Two weeks deadline no extension)
- 25% : Minor
- 45% : Major

Note: Grading will be relative. Score at least 38% to pass the course.

Total no. of lectures: 18

Book: Introduction to Algorithms by CLRS

What's a computational problem?

Given an input (aka instance) a comp. problem asks to return as output a solution satisfying some property.

Examples

Prime:

Input: A positive integer n .

Output: Yes/No depending on whether n is prime.

Sorting:

Input: A seq. of n numbers (a_1, a_2, \dots, a_n) .

Output: A permutation $(a'_1, a'_2, \dots, a'_n)$ of the input seq. s.t. $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

What's an algorithm?

An algo. is a seq. of computational steps that transforms the input to output.

An algo for a comp prob is correct, if for every input algo halts and gives the right answer.

Example: Algorithm to solve prime.

Prime (n):

If $n == 1$

return No

for $i = 2$ to $\frac{n}{2}$

If $n \% i == 0$

return No

return Yes

What is a data structure?

A data structure is a way to store and organise data in order to facilitate access & modifications.

Goals of this course:

- 1) Designing algorithms to solve computational problems.
- 2) Proving correctness of algorithms.

fun1(a,b) : \rightarrow gcd of a & b

If $b == 0$

return a

else

for $i = \text{max}(a, b)$ to 1

if $a \% i == 0$ and $b \% i == 0$
return i

gcd(a,b)

If $b == 0$

return a

else

return gcd(b, a%b)

}

Recursive

(15, 3) \rightarrow (3, 0)

$$\text{gcd}(a,b) = \text{gcd}(b, a \% b)$$

||

a % b

- 3) Argue about the efficiency of algorithms.

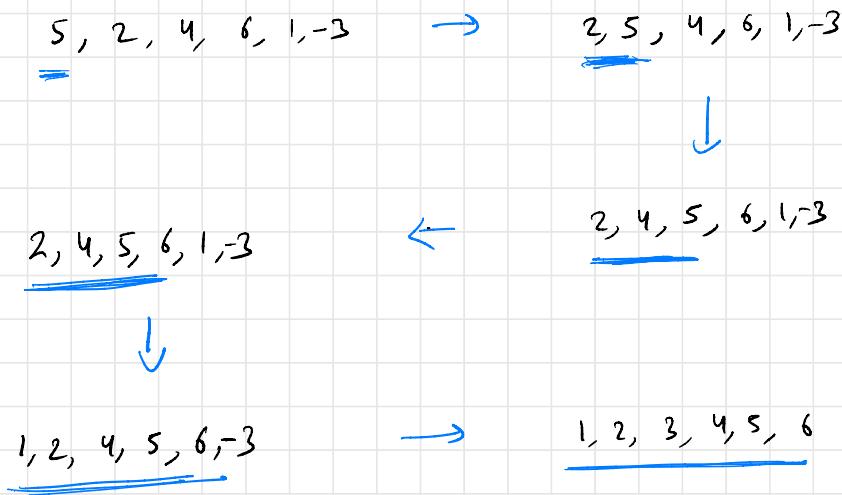
↑

will be mostly concerned with time -
sometimes space.

Insertion Sort

$$5, 2, 4, 6, 1, 3 \rightarrow 1, 2, 3, 4, 5, 6$$

Idea: To sort $A[1:n]$, we sort $A[1]$, then $A[1:2]$, then $A[1:3]$, ...



$A[1:6]$

$A[1:1], A[1:2], A[1:3], \dots, A[1:6]$

Pseudocode for Insertion sort:

Insertion-Sort (A, n):