

## Analysis of Algorithms

An **algorithm** is a step-by-step procedure for solving a problem in a **finite** amount of time.

### Running time

Most algorithms transform input data into output data. The running time of an algorithm typically grows with the input size. Average case time is often difficult to determine, so we (often) focus on the **worst case** running time. As it's easier to analyse

### Limitations of experiments

- It is necessary to implement the algorithm, which may be difficult or timeconsuming
- Results may not be indicative of the running time on other inputs not included in the experiment.
- In order to compare two algorithms directly, the same hardware and software environments must be used

### Limitations of theory

- It is necessary to implement the theory, which may be difficult or time-consuming
- Results may not be indicative of the typical running time on inputs encountered in real world.

### Theoretical analysis

A **high-level** description of the algorithm.

Characterizes running time as a function of the input size, **n**.

Allows us to evaluate the speed of an algorithm

**independently** of the hardware/software environment

```
Algorithm arrayMax(A, n)  
Input array A of n integers  
Output maximum element of A  
  
currentMax  $\leftarrow$  A[0]  
for i  $\leftarrow$  1 to n - 1 do  
    if A[i] > currentMax then  
        currentMax  $\leftarrow$  A[i]  
return currentMax
```