Algorithm Analysis Concepts

Algorithm analysis refers to the process of predicting the resources that the algorithm requires:

- memory
- communication bandwidth
- time - most often

Running time of an algorithm on a particular input is the number of primitive operations or steps executed.

The time taken by an algorithm depends on the input. In general – the time the algorithm takes grows with the size of the input data, so running time of an algorithm is described as a function of the input.
Size of input is usually denoted by $n$.

**Worst-case running time** – an upper bound on the running time for any input, in other words no matter what input the algorithm is given – it will not take longer than worst-case running time.

**Best-case running time** – shortest possible execution time of an algorithm; a lower bound on the running time for any input. Algorithm is guaranteed to take as long or longer than the best-case complexity on any input.

**Average-case running time** – how much time does the algorithm take for an “average” input.

We will make simplifying assumptions while evaluating the running time:

- ignore the actual cost of each operation, assume unit cost
- evaluate the rate of growth or order of growth and consider only the leading term of a formula describing running time,
- e.g. for running time $an^2 + bn + c$ - just use the leading term $an^2$ since the lower-order terms are relatively insignificant when $n$ is very large
- ignore the leading terms coefficient, since constant factors are less significant than the rate of growth, i.e. write that $an^2 = O(n^2)$. 