Chapter 1

1 What is an algorithm

An algorithm is a sequence of computational steps that transforms an input into an output, generally to solve a well-defined computational problem.

2 What is a data structure

Data structures are a way to store and organize data in memory to facilitate efficient access and modification (e.g., to enhance the speed of an algorithm).

3 How to quantitatively measure algorithm efficiency

Intuitively, it takes c units of time to perform a given computational operation. Typically the number of operations required by an algorithm corresponds to the size of the input n; therefore, algorithmic efficiency is expressed as a function of input size.

For instance, to sort n integers in increasing order, the *insertion sort* algorithm takes $c \cdot n^2$ units time, whereas the *merge sort* takes $c \cdot n \lg n$. Comparably speaking then, the $n \lg n$ algorithm will outperform the n^2 algorithm for large input sizes n.

There's an entire mathematical notation for identifying and comparing these input-efficiency functions for algorithms, called *asymptotic notation*; it's discussed at length in chapter 3. Below is a table of input sizes n which could be completed in time t for each efficiency function f_n, assuming each operation takes 1 ms.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$	2^{10^6}	2^{10^7}	2^{10^9}	$2^{10^{10}}$	$2^{10^{12}}$	$2^{10^{13}}$	$2^{10^{15}}$
$\sqrt{n}$	10^{12}	10^{15}	10^{19}	10^{21}	10^{24}	10^{26}	10^{30}
n	10^6	10^7	10^9	10^{10}	10^{12}	10^{13}	10^{15}
$n \lg n$	10^4	10^6	10^8	10^9	10^{10}	10^{11}	10^{13}
n^2	1000	7000	10^4	10^5	10^6	10^6	10^7
n^3	100	400	1500	4000	10^4	10^4	10^5
2^n	20	25	30	35	40	45	50
$n!$	9	11	12	13	15	16	17