

Introduction to algorithm - notes

Aheer Srabon

May 1, 2024

1 Divide and conquer

Divide the problem into one or more subproblems that are smaller instances of the same problem.

Conquer the subproblems by solving them recursively.

Combine the subproblem solutions to form a solution to the original problem.

1.1 Recurrences

A *recurrence* is an equation that describes a function in terms of its value on other, typically smaller, arguments.

- *Recursive case* - if a case involves the recursive invocation of the function on different (usually smaller) inputs.
- *Base case* - if it's not a recursive case.

The recurrence is *well-defined* if there is at least one function that satisfies it. *Ill-defined* otherwise.

A recurrence $T(n)$ is *algorithmic* if, for every sufficiently large threshold constant $n_0 > 0$, the following two properties hold:

1. For all $n < n_0$, we have $T(n) = \Theta(1)$.
2. For all $n \geq n_0$, every path of recursion terminates in a defined base case within a finite number of recursive invocations.