Algorithms: Binary Search Part-1 Pre-read

Relevel
by Unacademy

Asymptotic Analysis
Why needed?

It helps deduce mathematical relations denoting the Time Complexity of the Algorithm.
Using asymptotic analysis, we can get an idea about the algorithm's performance based on the input size. We should not calculate the exact running time but find the relation between the running time and the input size. We should follow the running time when the input size is increased.
For the space complexity, our goal is to get the relation or function of how much space in the main memory is occupied to complete the “algorithm”.

Asymptotic Behavior?

For a function f(n), the asymptotic behavior is the growth of f(n) as n gets large.
Small input values are not considered. Our task is to find how much time it will take for the large input value.
For example, f(n) = c * n + k as linear time complexity. f(n) = c * n2 + k is quadratic time complexity.
The analysis of algorithms can be divided into three different cases. The cases are as follows −
Best Case − Here, the lower bound of running time is calculated. It describes the the behavior of the algorithm under optimal conditions.
Average Case − In this case, we calculate the region between the upper and lower bound of the running time of algorithms. In this case, the number of executed operations is not minimum and not maximum.
Worst Case − In this case, we calculate the upper bound of the running time of algorithms. In this case, a maximum number of operations are executed.

Divide and Conquer

Divide and Conquer is an algorithmic paradigm where we have a problem in our hands. We try to divide the problem into smaller subproblems such that they are fine-grained into such a smaller level that it is easy to resolve those subproblems.
The resolution step is called the Conquer step. The combining step is to combine the answers from smaller subproblems to create the answer to the bigger problem. Binary Search comes in the class Divide and Conquer Algorithms.
It would help if you thought of a divide-and-conquer algorithm as having three parts:
1. Divide the problem into a number of subproblems that are smaller instances of the same problem.
2. Conquer the subproblems by solving them recursively. If they are small enough, solve the subproblems as base cases.
3. Combine the solutions to the subproblems into the solution for the original Problem.

Searching Algorithms

In computer science, the algorithms used to find or search an element(s) in a given data set are defined as searching algorithms. Searching algorithms are applied depending on the nature of the dataset. Here in this module, we will e focusing on two prime searching algorithms which are -

Linear search

Binary search.