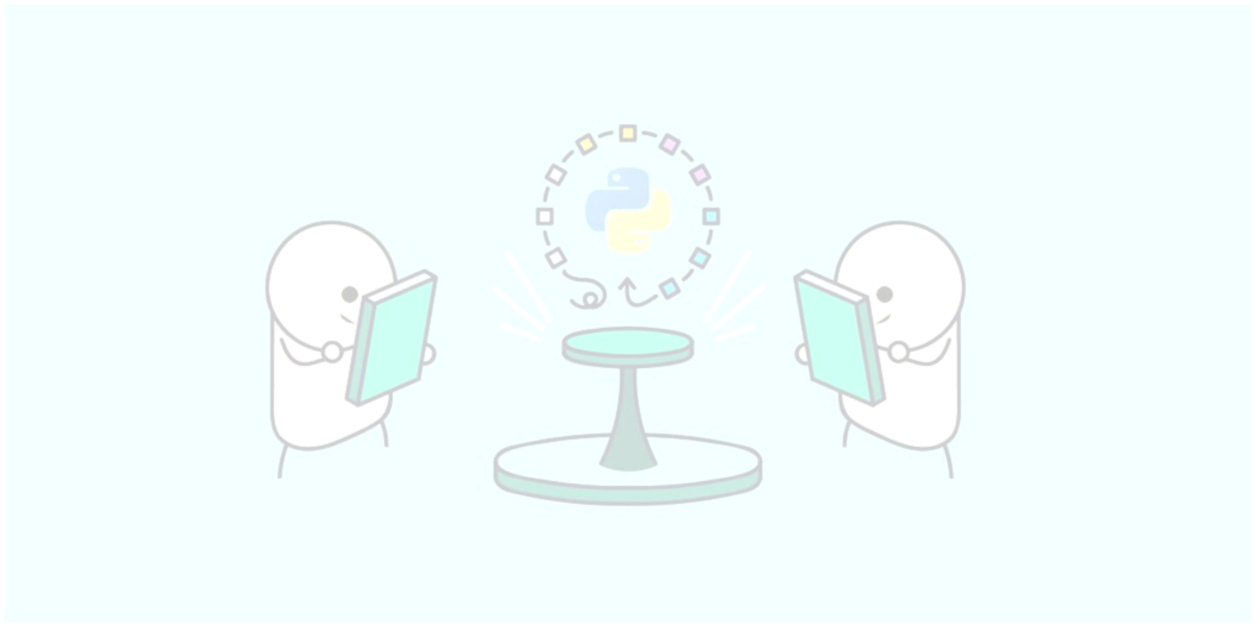
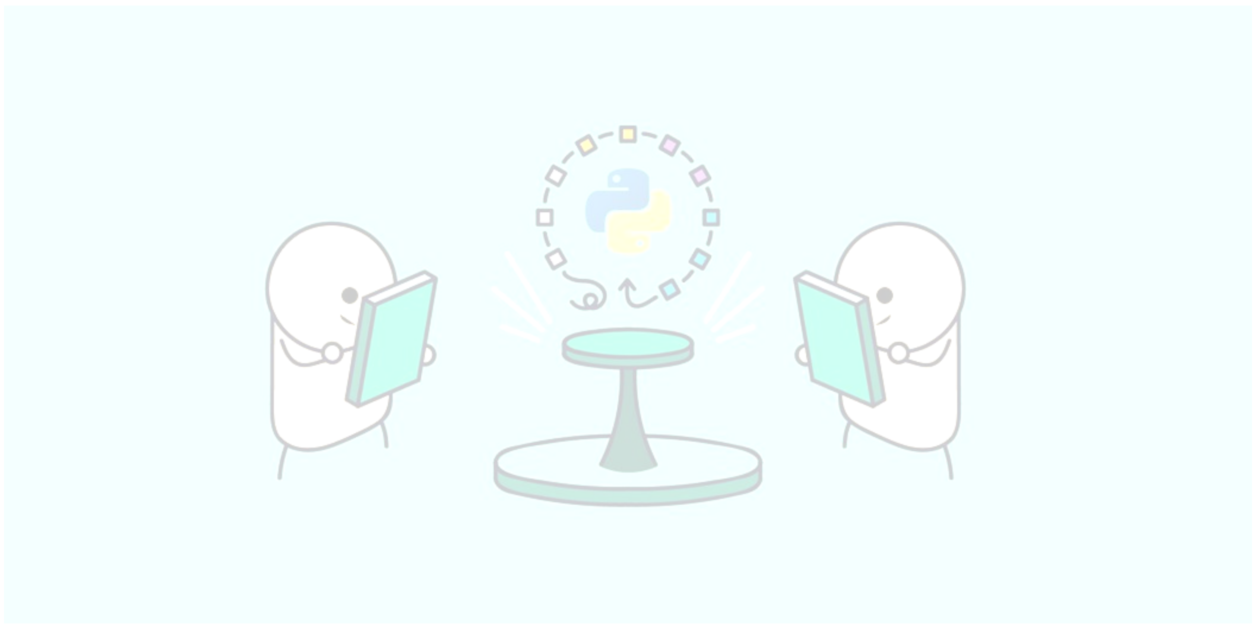
# Searching Algorithms

Rakin Mohammad Sifullah

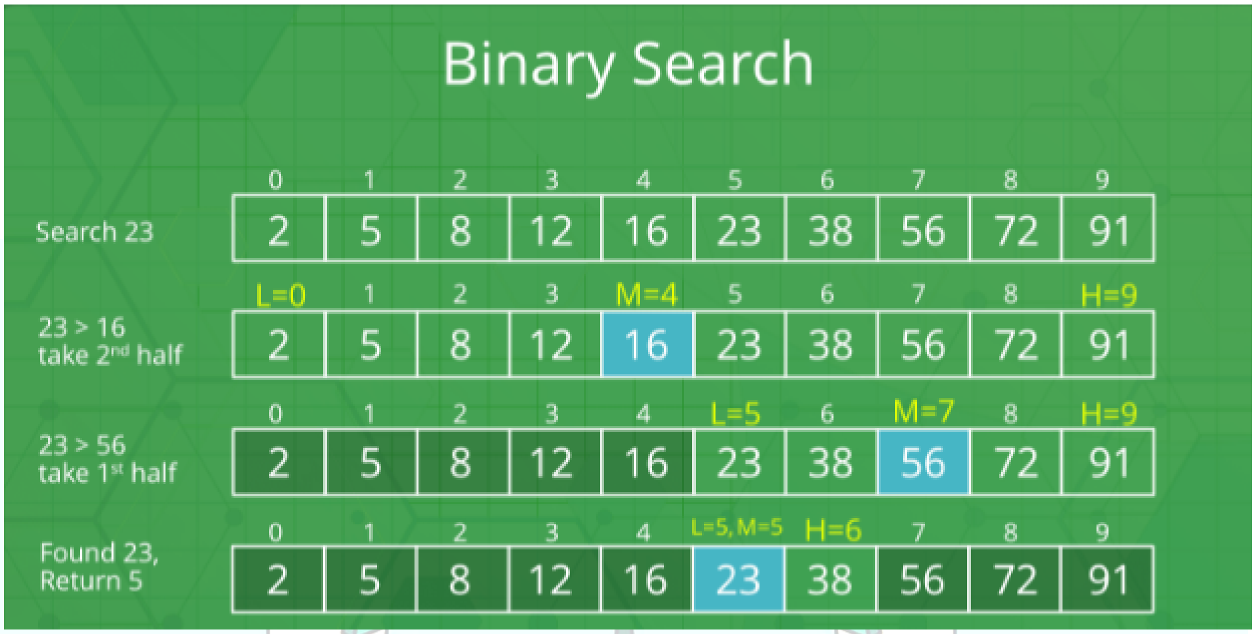
Mail : rakin.sifullah@gmail.com
GitHub : https://github.com/sifullahrakin





# Binary Search

Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise, narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.



The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n).
We basically ignore half of the elements just after one comparison.

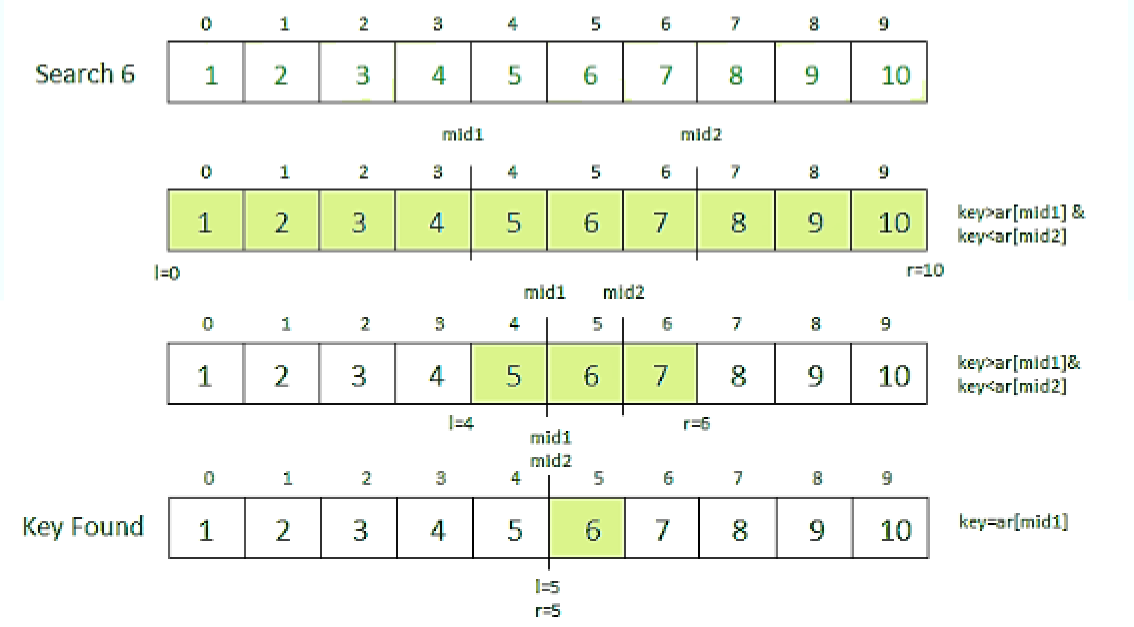
1. Compare x with the middle element.
2. If x matches with the middle element, we return the mid index.
3. Else If x is greater than the mid element, then x can only lie in the right half subarray after the mid element. So we recur for the right half.
4. Else (x is smaller) recur for the left half.

**Code Link :**
https://github.com/sifullahrakin/Searching-Algorithms/blob/main/Binary%20Search%20using%20Python

# Ternary Search

A ternary search algorithm is a technique in computer science for finding the minimum or maximum of a unimodal function. A ternary search determines either that the minimum or maximum cannot be in the first third of the domain or that it cannot be in the last third of the domain, then repeats on the remaining two thirds. A ternary search is an example of a divide and conquer algorithm. Here the array need not be sorted.

- First, we compare the key with the element at mid1. If found equal, we return mid1.
- If not, then we compare the key with the element at mid2. If found equal, we return mid2.
- If not, then we check whether the key is less than the element at mid1. If yes, then recur to the first part.
- If not, then we check whether the key is greater than the element at mid2. If yes, then recur to the third part.
- If not, then we recur to the second (middle) part.
- Ternary search can be implemented in recursive and iterative ways.



**Code Link :**
[https://github.com/sifullahrakin/Searching-Algorithms/blob/main/Ternary%20Search%20using%20python](https://github.com/sifullahrakin/Searching-Algorithms/blob/main/Ternary%20Search%20using%20python)

## *References :*

1. ***Data Structures & Algorithms in Python*** by Goodrich, Tamassia and Goldwasser
2. ***Programming contest, Data Structure and Algorithm*** by MD. Mahbubul Hasan
3. ***https://www.geeksforgeeks.org/***