Binary Search:

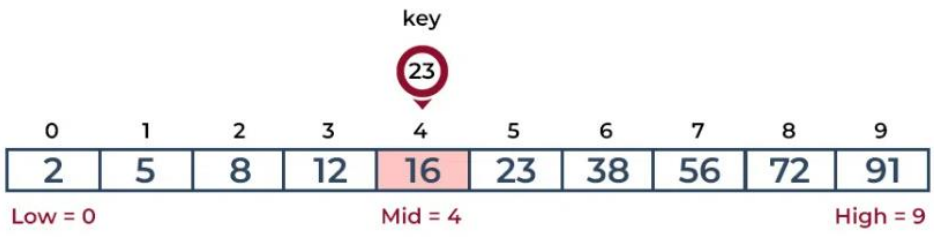
Binary Search is defined as a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(log N).

How does Binary Search work?

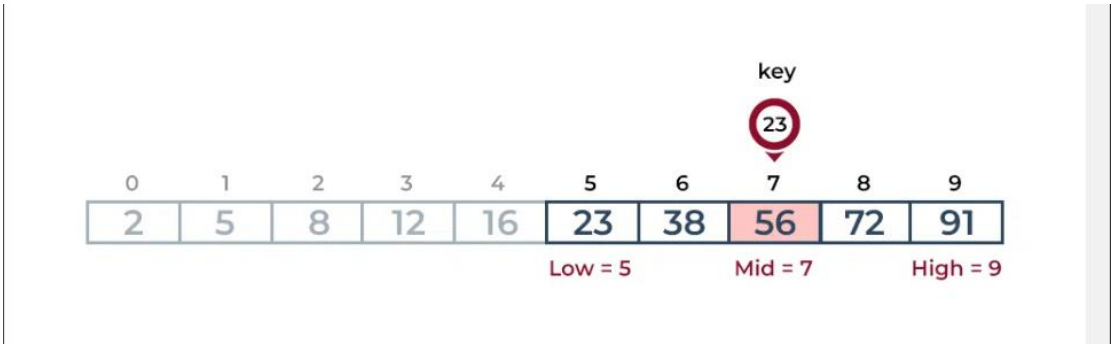
Consider an array arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}, and the target = 23.

First Step: Calculate the mid and compare the mid element with the key. If the key is less than mid element, move to left and if it is greater than the mid then move search space to the right.

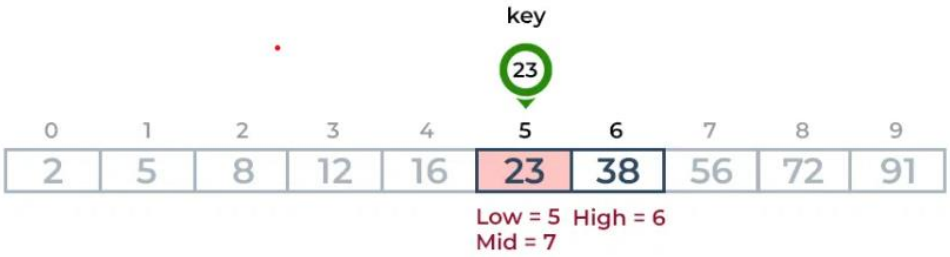
- Key (i.e., 23) is greater than current mid element (i.e., 16). The search space moves to the right.



- Key is less than the current mid 56. The search space moves to the left.



Second Step: If the key matches the value of the mid element, the element is found and stop search.



Strassen's Matrix Multiplication

Strassen's Matrix Multiplication is the divide and conquer approach to solve the matrix multiplication problems. The usual matrix multiplication method multiplies each row with each column to achieve the product matrix. The time complexity taken by this approach is O(n3), since it takes two loops to multiply. Strassen's method was introduced to reduce the time complexity from O(n3) to O(nlog 7).

Naïve Method

First, we will discuss naïve method and its complexity. Here, we are calculating Z=***X***X × Y. Using Naïve method, two matrices (X and Y) can be multiplied if the order of these matrices are p × q and q × r and the resultant matrix will be of order p × r. The following pseudocode describes the naïve multiplication −

Algorithm: Matrix-Multiplication (X, Y, Z)
for i = 1 to p do

```
  for j = 1 to r do
    Z[i,j] := 0
    for k = 1 to q do
      Z[i,j] := Z[i,j] + X[i,k] × Y[k,j]
```

Complexity

Here, we assume that integer operations take O(1) time. There are three for loops in this algorithm and one is nested in other. Hence, the algorithm takes O(n3) time to execute.

Strassen's Matrix Multiplication Algorithm

In this context, using Strassen's Matrix multiplication algorithm, the time consumption can be improved a little bit.

Strassen's Matrix multiplication can be performed only on square matrices where n is a power of 2. Order of both of the matrices are n × n.

Divide X, Y and Z into four (n/2)×(n/2) matrices as represented below −

Z=[IKJL]
X=[ACBD]
and Y=[EGFH]

Using Strassen's Algorithm compute the following −

M1:=(A+C)×(E+F)
M2:=(B+D)×(G+H)
M3:=(A−D)×(E+H)
M4:=A×(F−H)
M5:=(C+D)×(E)
M6:=(A+B)×(H)
M7:=D×(G−E)
Then,

I:=M2+M3−M6−M7
J:=M4+M6
K:=M5+M7
L:=M1−M3−M4−M5

Analysis

T(n)={c7xT(n2)+dxn2ifn=1otherwisewherecanddareconstants

Using this recurrence relation, we get T(n)=O(nlog7)

Hence, the complexity of Strassen's matrix multiplication algorithm is O(nlog7)