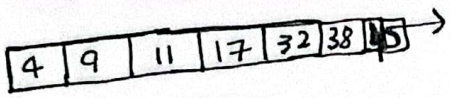
Algorithms

(1) Binary Search :

Consider an array :



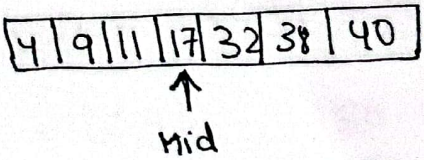
If you want to search 32 you will go one by one from start by comparing numbers and in end you will get 32 at index 4.

→ Consider a biggy example where we have 30,000 elements or records, if we go linearly with Time complexity of O(n) - It is not efficient.

↳ so in above examples linear search will take 4 iterations to get 32.

→ Binary Search works on sorted list.

↳ In binary seach we will take middle element of array.



→ Now as we are searching 32 so 32 > 17 so we will rule out left side of array.

→ so we got:

32	38	40

⇒ Take Middle again and compare with 32.

⇒ As 38 > 32 so it will be on left side so discard right side. ie Mid.

↳ 32

⇒ NOW compare the last element & you got 32.

→ So you did this in 3 iteration. In records with large elements it makes big difference.

So every iteration you are doing $n \to n/2$

iteration 1 ⇒ $n/2$

iteration 2 ⇒ $n/2/2 = n/2^2$

" 3 ⇒ $n/2^2/2 = n/2^3$

⋮

iteration K ⇒ $n/2^K$

$1 = n/2^K$

$n = 2^K$

$\log_2 n = \log_2 2^K \Rightarrow \log_2 n = K \underbrace{\log_2 2}_{1}$

$K = \log_2 n$ ⇒ So Time complexity of Binary search in $O(\log n)$