Introduction to Algorithms :→

Algos

Searching (an Item)
- * Linear Search *
- * Binary Search *
- * Recursive Binary Search *
- * Jump Search
- * Interpolation Search

Sorting (Data)
- * Bubble Sort
- * Insertion Sort
- * Selection Sort
- * Merge Sort
- * Quick Sort
- * Heap Sort

Comparison Algos

NCA :
- * Count Sort
- * Radix Sort

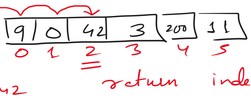
Shell Sort

Linear Search :→

Element == Target key

Normal Traversal Search

Big O Notation



syntax

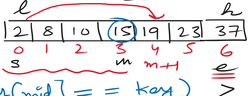
key = 42

return index pos

```
O(1) if( arr[i] == target )
O(n)     return i ;
```

Binary Search : Pre-Condition Sorted Array

key = 23



s, e, m, m+1

mid = (s+e)/2

(0+6)/2

(0+2)/2 = 1

① if (arr[mid] == key) return mid ;

② if (arr[mid] < key) s = m+1 ; → go to right side

15 < 23

③ if arr[mid] > key end = m-1 ; (15 > 8) shift to left

s ≤ e

e = em = ③

m = 3

key = 8

arr[mid] == key

DRY - RUN (Mercedes) (Pure Storage)

Binary Search Time Complexity

Array / Sorted

Initial Size N

Left part | right part

$\frac{N}{2^0} = 1$ mid = (s+e)/2

$\frac{N}{2}$ $\frac{N}{2}$ = $\frac{N}{2^1}$

$\frac{N}{2^2}$ $\frac{N}{2^2}$ $\frac{N}{2^2}$

$\frac{N}{2^k} = 0, 1, 2, 3 \ldots$

$k = \log_2 N$

Big O log N

$\frac{N}{2^0}$ $\frac{N}{2^1}$ $\frac{N}{2^2}$ $\frac{N}{2^3}$ $\frac{N}{2^k}$

N N/2 N/4 N/8 ...

where k = 0, 1, 2, 3 . . .

$k = \log_2 N$

Big O Notation log N

Optimal Mid Value Formula :→

Integer : short int long

−128 to 127

INT_MAX → $2^{31} - 1$

INT_MIN → -2^{31}

$m = \frac{s+e}{2}$

$= \frac{(INT\_MAX + INT\_MAX)}{2}$ → OOB TLE

$\frac{s+e}{2} = s + \frac{[e-s]}{2}$

$= \frac{s}{1} + \frac{e-s}{2} = \frac{2s+e-s}{2}$

$m = s + \frac{(e-s)}{2}$ ✓

$= \frac{s+e}{2}$

Important Binary Search Questions for Placements :

Leet Code / Coding Ninjas

- * Search In A 2D Matrix
- * Square Root of a given Number
- * Aggressive Cows
- * Book Allocation Problem
- * Peak In A Mountain
- * First / Last / Total Occurrences of an element