Kiran Kumar 11/12/23

DSA

Date
Page

☆ Searching Algorithms

- Sequential Search Algo
- Binary Search Algo

⇒ **Sequential Search Algorithm**

⇒ Straight-forward algorithm.

⇒ Searches the given key element in the list of n elements by checking successive elements in the list until a match is found or list is exhausted.

⇒ Also called Linear Search.

☆ Algorithm SequentialSearch $[A(0 \ldots n-1), K]$

// searches for a given value in a given array by sequential search

// input → the array $A[0 \ldots n-1]$ and a search key K.

// output → The index of the first element in A that matches K or -1 if there are no matching elements.

```
i ← 0
While i<n and a[i] != K do
        i ← i+1
if i<n
    return i
else
    return -1.
```

★

| a[0] | a[1] | a[2] | a[3] | a[4] |
|---|---|---|---|---|
| 15 | 12 | 7 | 10 | 8 |

n = 5

K = 7 → searching element.

iteration 1 → i = 0

while 0 < 5 and 15 != 7 → true.
a[0]

i = 0 + 1 = 1

iteration 2 → i = 1

while 1 < 5 and 12 != 7 ⇒ true.
a[1]

i = 1 + 1 = 2

iteration 3 → i = 2

while 2 < 5 and 7 != 7 ⇒ False.
a[2]

2 < 5 → true (true) and 7 != 7 (False)

return i → a[i] → a[2]

⇒ (7) → is returned

⇒ Time Complexity

↳ best Case ⇒. Search key is 1st element only one iteration is required →

$$C_{best}(n) = 1$$

↳ Average case efficiency ⇒ .

$$C_{avg}(n) = \frac{n+1}{2} = \frac{5+1}{2} = \frac{6}{2} = \left(\frac{3}{2}\right)$$

comparisons

↳ Worst case ⇒

$$C_{worst}(n) = \underbrace{n}_{\text{size of array}}.$$

★ Binary Search ⇒ array should be sorted

↳.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| A | 2 | 6 | 13 | 21 | 36 | 47 | 63 | 81 | 97 |

$$mid = \frac{8}{2} = 4 \qquad \left[\frac{start + end}{2}\right] \quad \begin{matrix} 0 & & 8 \\ & \searrow & \swarrow \\ & 4 & \end{matrix}$$

a[4] ↓

a[mid] = 36.

Search key = 21.

a[mid] = 36 — 36 > 21. ⇒ Search Left side.

a[5] = 21

| 0 | 6 | 13 | 21 | 36 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

$mid = \frac{0+4}{2} = 2$

$a[2] = 13$

$13 < 21 \Rightarrow$ Search Right side

| 13 | 21 | 36 |
|---|---|---|
| 2 | 3 | 4 |

$mid = \frac{2+4}{2} = \frac{6}{2} = 3$

$a[3] = 21$

$21 = 21 \rightarrow$ Search successful.

---

Algorithm Binary Search } $\Rightarrow$ follows divide & conquer Strategy.

$l \leftarrow 0$ ; $r \leftarrow n-1$

While $l \leq r$ do

$m \leftarrow [(l+r)/2]$

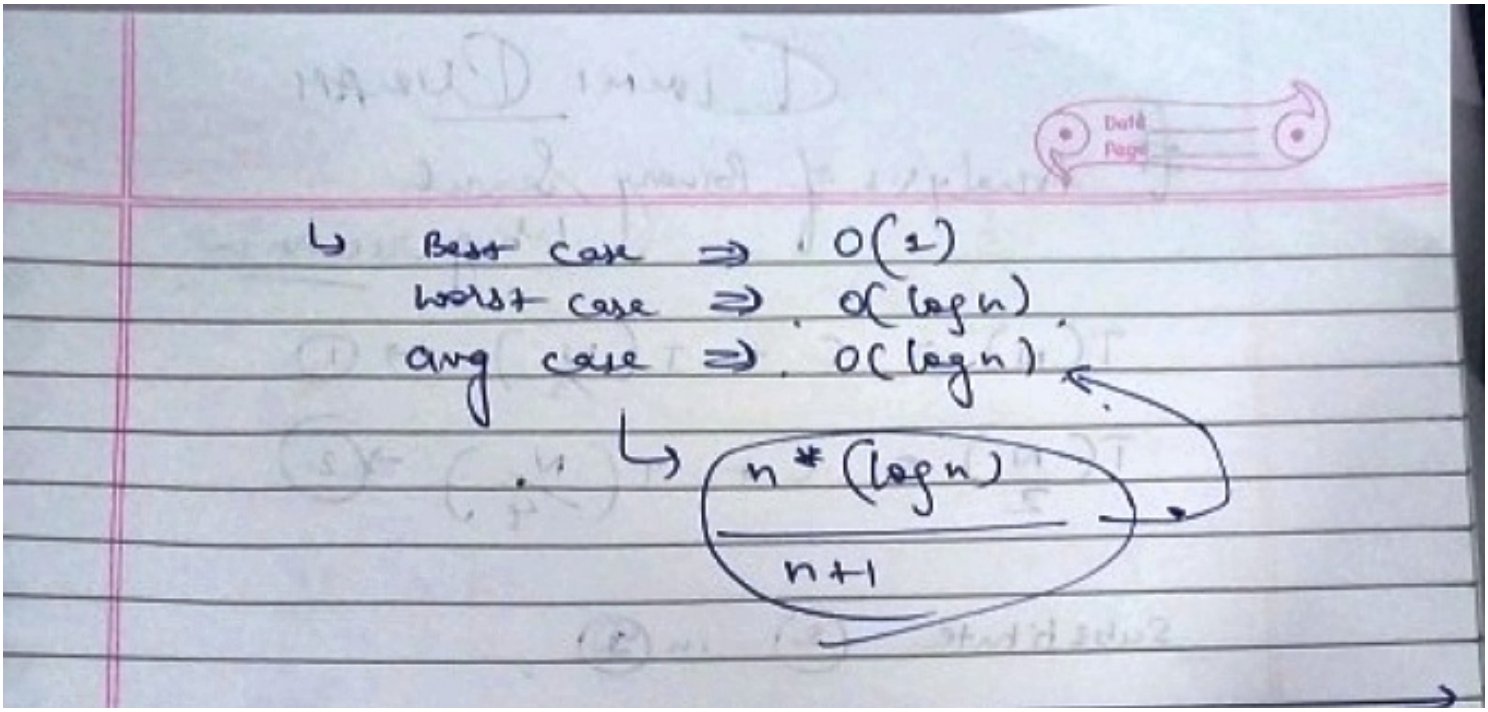
if $K = A[m]$ return $m$

else if $K < A[M]$ $r \leftarrow m-1$

else $l \leftarrow M+1$

return $-1$

$\Rightarrow$ ...


Best case ⇒ O(1)
Worst case ⇒ O(log n)
avg case ⇒ O(log n)
n * (log n)
n+1