* Binary Search Applications : Coding Interview Questions

TCS / Capgemini / Wipro / IBM / Intel / HCL / DXC

* Square root of a number (Integral Part) using Binary Search : [log N]

(1, 2, 3, 4, 5) key = 2
index 1 to 5

int → n = 36 → o/p = 6
n = 37 → o/p = 6
n = 65 → o/p = 8

* Search Space

n = 36 → ⇒ 0 — 36
s e

$\sqrt{36}$

0 — 17

ans = -1
ans = mid (3)
ans = mid (5)

0 — 7 $\frac{17}{2}$ = 8

4 — 7 $\frac{7}{2}$ = 3

$\frac{11}{2}$ = 5

8×8 = 64
e = m-1
e = 8-1
= 7

3×3 = 9
9 < 36
s = m+1
s = 3+1 = 4

6 — 7

$\frac{6+7}{2} = \frac{13}{2}$ = 6

5×5 = 25
s = m+1
s = 5+1
= 6

6×6 = 36

mid = $\frac{0+36}{2}$
= 18
18×18 = 324
e = m-1
18×18 > n
e = 18-1
= 17

* First, Last & Total Occurrences of an element in a given sorted array in LogN time complexity.

arr[3] == (3) ?

Key (3)

ans = -1 ;

| 1 | 2 | 3 | 3 | 3 | 4 | 5 |
0 1 2 3 4 5 6

= ✓ FO = (2)
✓ LO = (4)

⇒ TO = LO - FO + 1
= 4 - 2 + 1
= 2 + 1
= 3

mid = $\frac{s+e}{2} = \frac{6}{2}$ = 3

```
if (arr[mid] == key) {
    ans = mid;
    e = m - 1; (left)
}
else if (            ){
}
else {
}
```

```
if (arr[mid] == key) {
    ans = mid;
    s = m + 1; (right)
}
else if (            ) {
}
else {
}
```

Kunal Kushwaha (Striver / Babbar) Abdul Bari

* Search in a 2D Matrix
* Find the missing number in an array using Binary Search.
* Peak in a mountain / Peak Index In An Array.
* Sorted array to Balanced Binary Search Tree.
* Book Allocation Problem
* Aggressive Cows.
* Painter's Partition Problem.

✓ LeetCode
✓ Coding Ninjas
✓ Code Forces

These are all Binary Search Applications.

Selection Sort (n^2)

* [Swap]

| 1 | 2 | 4 | 6 | 8 | 9 |
0 1 2 (3) 4 5
i i j

j = i+1

min Index = i
min Index = j

(n-1) for
for

arr[i] and arr[minIndex]
Swap

Insertion Sort (n^2)

* [Shift] (Uno)

While

temp = 2

| 9 | 2 | 6 | 1 | 4 | 5 |

temp

| 2 | 9 | 6 | 1 | 4 | 5 |

| 2 | 6 | 9 | 1 | 4 | 5 |

| 1 | 2 | 6 | 9 | 4 | 5 |

* Given an array of only 0's, 1's & 2's Sort the array without using any sorting algorithm in O(n) Time complexity.

I/p [1, 2, 1, 2, 0, 2, 1]

O/p [0, 1, 1, 1, 2, 2, 2]

```
for (datatype var : collection) {
    statements
}
```

c0 = 1-1=0 0 1 1 1 2 2 2
c1 = 3-1=2-1-1-1=0
c2 = 3 0 1 1 1 2 2 2

Jagged Arrays