* Sort 0's 1's 2's :

Given an array of only 0's , 1's & 2's sort the given array without using any sorting algorithm.

i/p → arr = [1, 2, 0, 1, 2, 0]

o/p → arr = [0, 0, 1, 1, 2, 2]

c0 = 2-1-1 c1 = 2-1-1 c2 = 2-1-1

index = 0

arr [0 0 1 1 2 2]

* Binary Search Algo: →   key = 19 → index   key = 6

Sorted Array

2	6	8	10	12	15	19	21
0	1	2	3	4	5	6	7

2	9	4	6	1
0	1	2	3	4

key = 6

return 3

return -1

O(n)

O(1)

s = m+1

e = m-1

(i) arr[mid] == key
return mid;

(ii) else if arr[mid] < key
10 < 19
s = m+1 = 4

(iii) else (arr[mid] > key) 10 > 6
e = m - 1

mid = $\frac{s+e}{2}$ = $\frac{7}{2}$ = 3

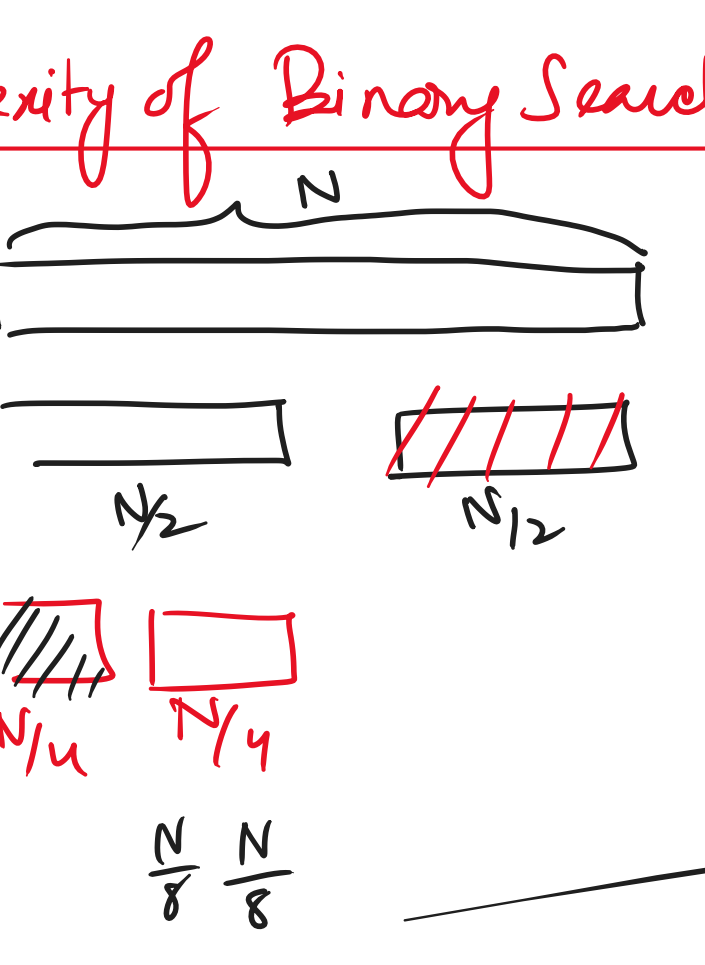
4-7

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

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

6/7

Time Complexity of Binary Search: →



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

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

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

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

↓ constant

mid

$k = \log_2 N$

Edge cases or corner cases for Binary Search:

* Only 1 element in the array
The start & end will coincide at a single index, index = 0.

* The start value will never be greater than end value.
(s <= e)

* Optimal formula for finding the mid in Binary Search: → $(2^{31}-1)$ (-2^{31})

(s = MAX   e = MAX) index

→ m = $\frac{s+e}{2}$ -ve   OOB

(-ve)

$m = s + \frac{e-s}{2}$ → This eliminates the boundary condition

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

* Binary Search Coding Questions for Placements:

(i) Square root of a number (integral part).
(ii) Search in a 2D Matrix.
(iii) First, Last & Total occurrence in array.
(iv) Missing element in an array.
(v) Aggressive Cows
(vi) Book Allocation Problem
(vii) Painter's Partition Problem
(viii) Peak In a Mountain

} log N

* Square root of a number using Binary Search

n = 36   √36 → 0 — 36   mid = $\frac{36}{2}$ = 18

x × x = n   37-6 35-5

ans = 0
ans = mid (3) (5)

1, 2

5×5 = 25

6×6 = 36

e = m-1
0 — 17
e = m-1
0 — 7
s = m+1
4 — 7
s = m+1 = 5
6 — 7

$\frac{13}{2}$ = 6 $\frac{11}{2}$

18×18
324 > n
e = m-1
$\frac{17}{2}$ = 8
8×8 = 64
$\frac{7}{2}$ = 3
3×3 = 9

n = 64

ans = 7   3×3   5×5   6×6

s e
8 9 S = m+1

ans = m

$\frac{17}{2}$

= m = 8

8×8 = 64

0 — 64
e = m-1
0 — 31
e = m-1
0 — 14
s = m+1
8 — 14
e = m-1
8 — 10
8 — 9

$\frac{17}{2}$ = 8 $\frac{18}{2}$ = 9

32
32×32 = 1024
$\frac{31}{2}$ = 15
225
$\frac{14}{2}$ = 7
$\frac{22}{2}$ = 11
121

First, Last, Total Occurrences of an element   total = lo − fo + 1   (3)

(log n)

[key = 3]

0	1	2	3	4	5	6	7
1	2	3	3	3	4	5	6

s   mid   e

mid = $\frac{7}{2}$ = 3

ans = -1

if (arr[mid] == key) {
→ ans = mid;
→ e = m-1; Left
}
else if (arr[mid] < key)
s = m+1;
else e = m-1;

fo = 2

if (arr[mid] == key)
→ ans = mid;
Right → s = m+1;
else if (arr[mid] < key)
s = m+1;
else e = m-1;

lo = 4

* Cognizant → Sep 2024 | Dec 2024 | Feb 2025

Find the missing element in a sorted array using log N time complexity.   6LPA 18LPA 12LPA

1 — N

0+1	1+1	2+1	3+1	x	x	x
1	2	3	4	6	7	8
0	1	2	3	4	5	6

s   e

[element = index+1]
correct

$\frac{6}{2}$ = 3

if (arr[mid] == mid+1)
arr[3] == 3+1
4 == 4
s = m+1;
arr[5] != 5+1
7 != 6
e = m-1

* We came to the left
mid = 4

s = 4, e = 4

if arr[mid-1] == mid
arr[3] == 4
4 == 4

0+1 = 1   (4)   4+1 = 5   return (m+1)

while going to left:

if (arr[mid-1] == mid || mid == 0)
return m+1;   (5)   (1)   0+1

(arr.len + 1)

length = 7 of 8

1, 2, 3, 4, 5, 6, 7

(7+1)

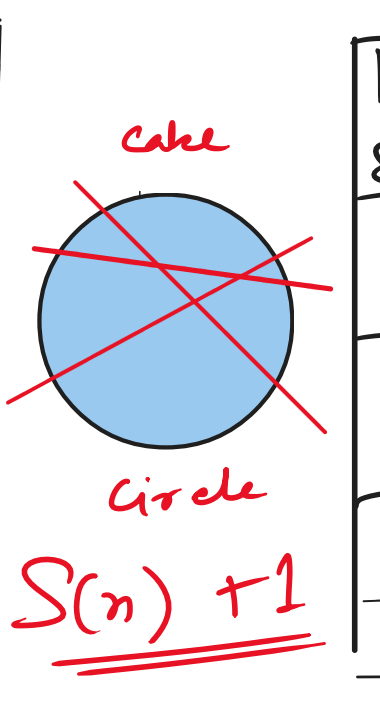
* Baker's Cake Pieces Problem   Cognizant | TCS | Wipro

or

Max Regions In A Plane With N Straight Lines

Number of Cuts	Max Pieces
S(0) + 1	1
1 + 1	2
2	4
3	7
4	11

cake



Circle

S(n) + 1

Number of Straight lines	Max Regions
0	1
1	2
2	4
3	7
4	11

* What we call functions in C are called methods in Java.

Syntax of methods in java: →

return_type method_name ( ) {   You can add parameters to the method here.

public static void sum(int a)   → parameters optional

}

→ Many ways to declare a method: →

(1) access modifier static returntype method (){

access modifier: optional   static: optional   returntype: mandatory

}

Access Modifier Table   (folder)

Modifier Name	Inside Class	Outside Class	Inside Package	Outside Package
public	Yes	Yes	Yes	Yes
private	Yes	No	No	No
protected	Yes	Yes	Yes	Yes (Inheritance)
default (P2)	Yes	Yes	Yes	NO