Tasks/HW

21 May 2020 09:32

Tasks - May 20
Level - 1 Questions (Array)*
https://practice.geeksforgeeks.org/problems/sum-of-array-elements/0
https://practice.geeksforgeeks.org/problems/print-alternate-elements-of-an-array/1
https://practice.geeksforgeeks.org/problems/print-the-left-element/0
https://practice.geeksforgeeks.org/problems/check-if-an-array-is-sorted/0
https://practice.geeksforgeeks.org/problems/find-triplets-with-zero-sum/1

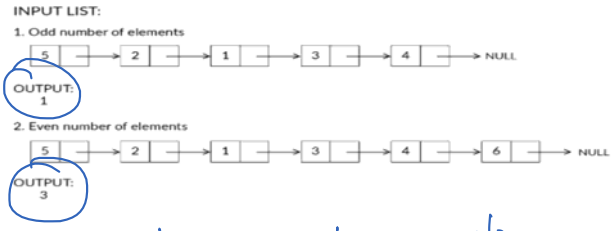
Level - 2 Questions (Array)*
https://practice.geeksforgeeks.org/problems/fascinating-number/0
https://practice.geeksforgeeks.org/problems/rotate-array-by-n-elements/0
https://practice.geeksforgeeks.org/problems/sum-of-fai-aj-over-all-pairs-in-an-array-of-n-integers/0
https://practice.geeksforgeeks.org/problems/count-pair-sum/0
https://practice.geeksforgeeks.org/problems/segregate-even-and-odd-numbers/0

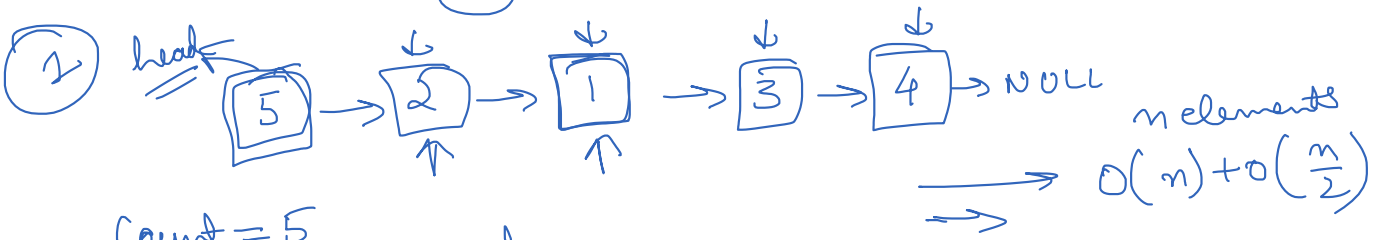
Reading:
Read all these topic from www.realjavaonline.com

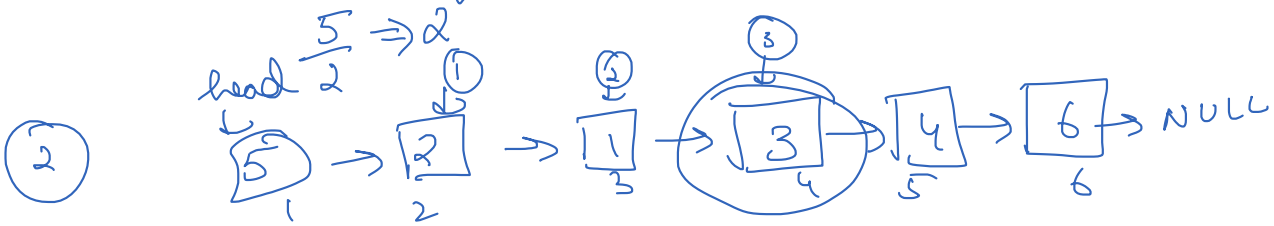
I suggest that you guys buy this book, whosoever is interested can buy: https://amzn.to/2WNK88j

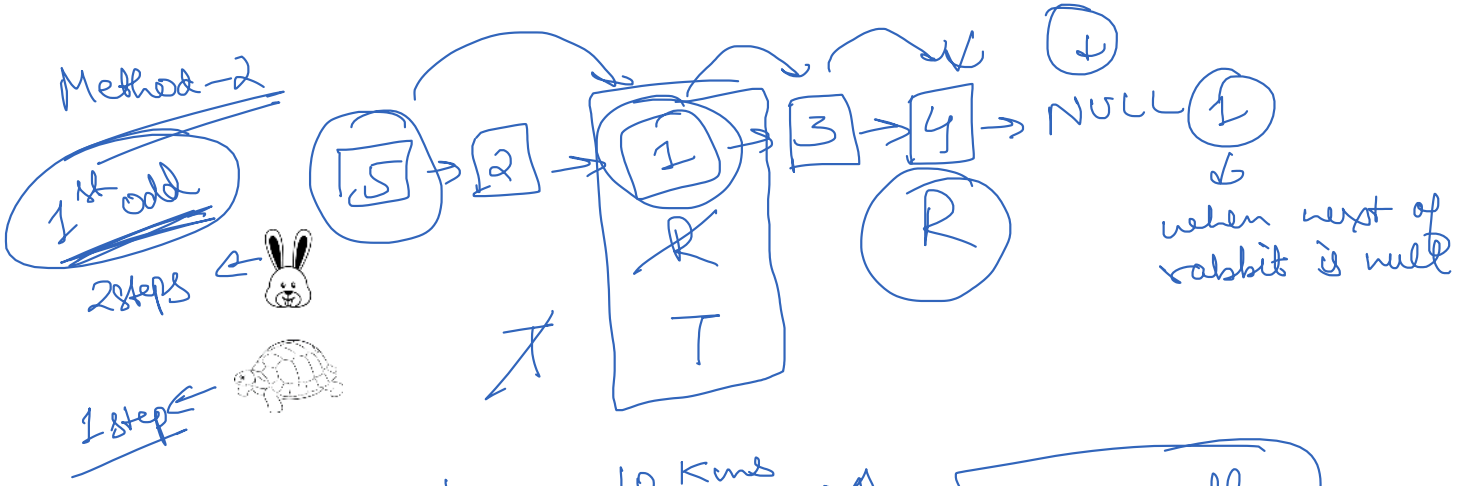
ID - Middle Element LL

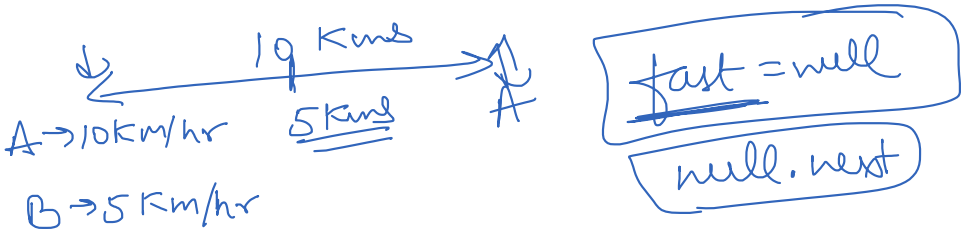
22 April 2020 22:24

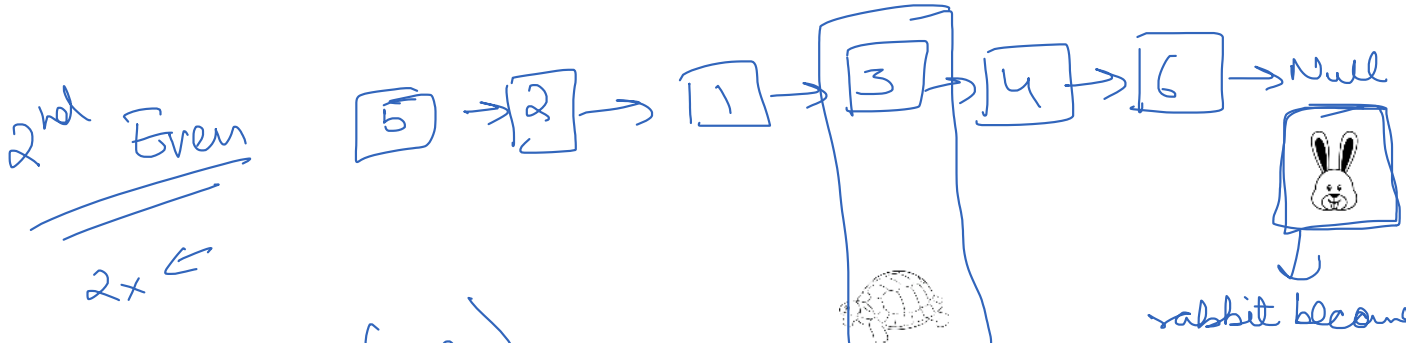








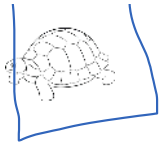




2x ←

x ←

$O\left(\frac{n}{2}\right)$



rabbit becomes null

Visible?

int 1, 2, 3, 4, 5

Abhinav, Ananya, Divya ___

DS → Arrays

↳ Indexed Collection

↳ Finite no.

↳ homogenous Elements

Array is an object

arr [] → naming

int x → declaration

x = 5; → initialization

int arr[]; → Declaration

or

int []arr;

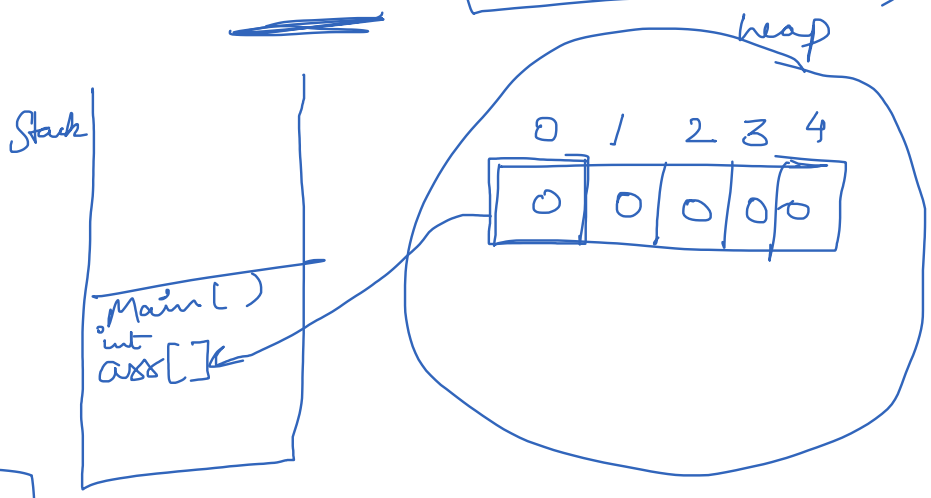
Creating an Array

(a) Array Creation Expression

(b) Array Initializer List

a → arr = new int [5];

heap



arr[0];
arr[1];

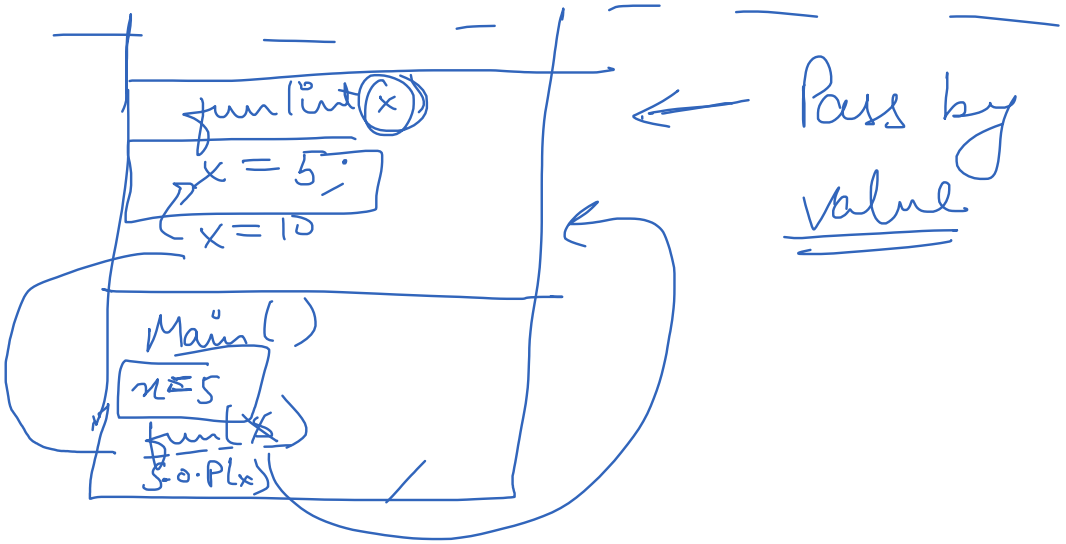
int = 4 bytes
Array → 20 bytes

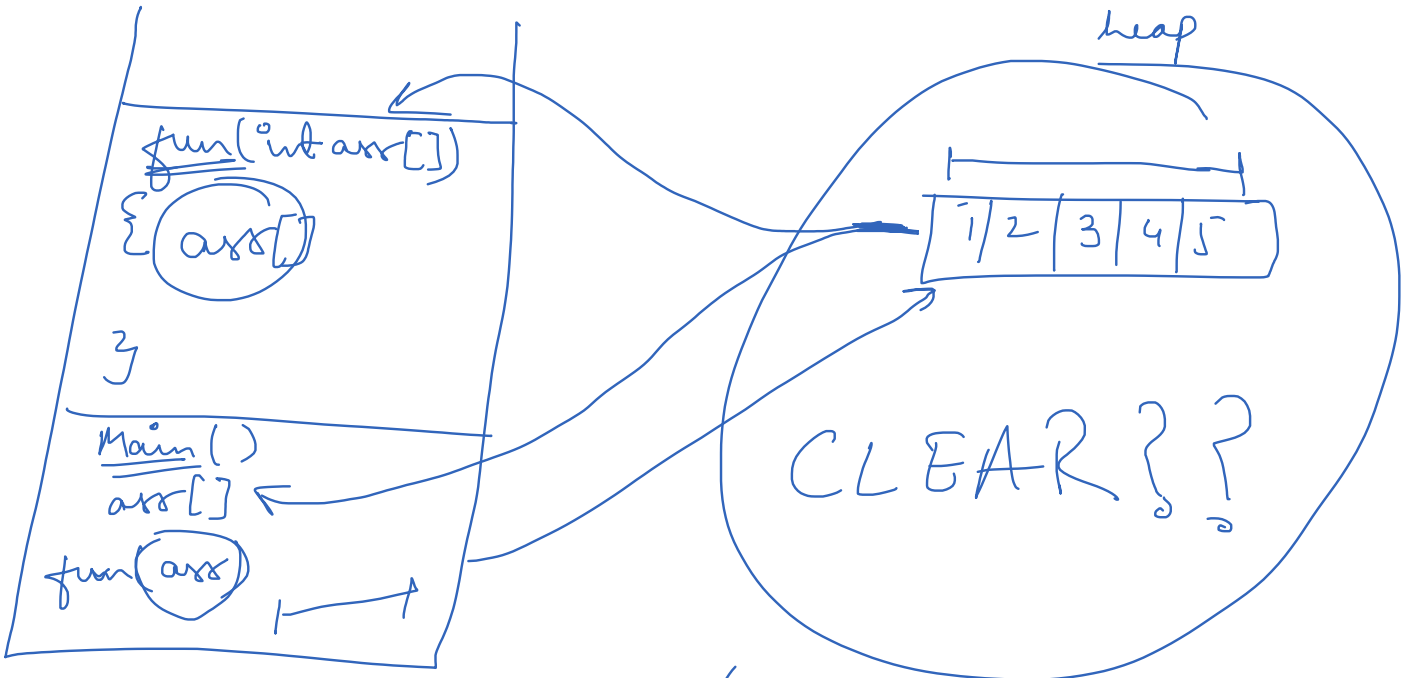
n element array → [0] to [n-1]

```
int arr[] = new int[10];
```

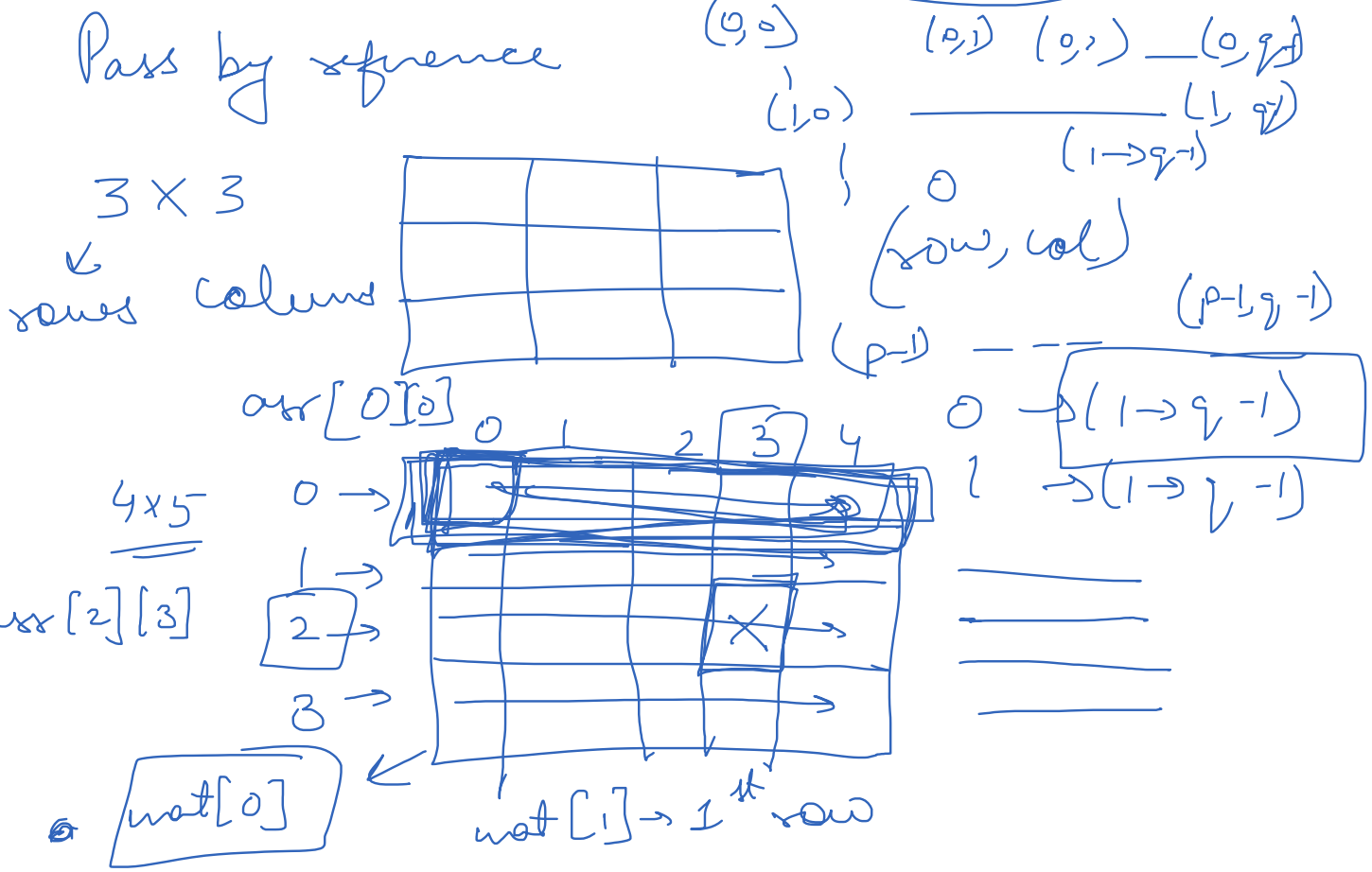
(b) Initializers list

```
int arr[] = {4, 5, 6, 7, 8};
```





Pass by reference

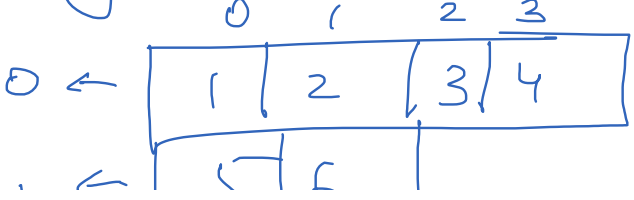


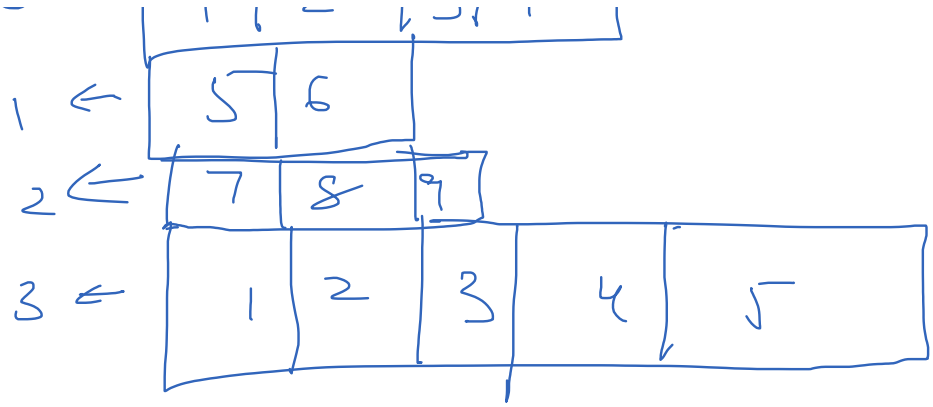
mat

mat.length → 4

Jagged Array

mat[i].length





(2) Ways

(a) int A[][] = {
{1, 2, 3, 4}
{5, 6}
{7, 8, 9}
{1, 2, 3, 4, 5}
}

(b) int A[][] = new int [4][];

(*) A[0] = {1, 2, 3, 4};
A[1] = new int [2];
A[2] = new int [3];
A[3] = new int [5];

May - 21

21 May 2020 19:58

For Loop I
Time Complexity (Prime Example)
Sorting
Recursion

→ Importance, How to check

Prime no. check

10

Ram	Shaam
n → [2 to n-1]	n → [2 to √n]
divide by every no.	
$\frac{n}{2}, \frac{n}{3} \dots \frac{n}{n-1}$	
% → 0 ✗ Prime, % → 0 Not Prime	
n → computations	√n computations

16 → √. → 4

(1 computation = 1 unit time)

n	Ram	Shaam
n=10	10	3
n=100	100	10
n=10,000	10,000	100

← Efficient

10 unit
10 sec
20 unit
20 sec

10 unit
20 sec
20 unit
40 sec

Time Complexity

1. Atomic/Basic Statements
   - ↳ Unit time
   - x + 1 → Addition/Subtraction
   - x == 0 → true / false
   - x = 5 → Assignment
   - S.O.P → Output

x = x+1 → 2 units

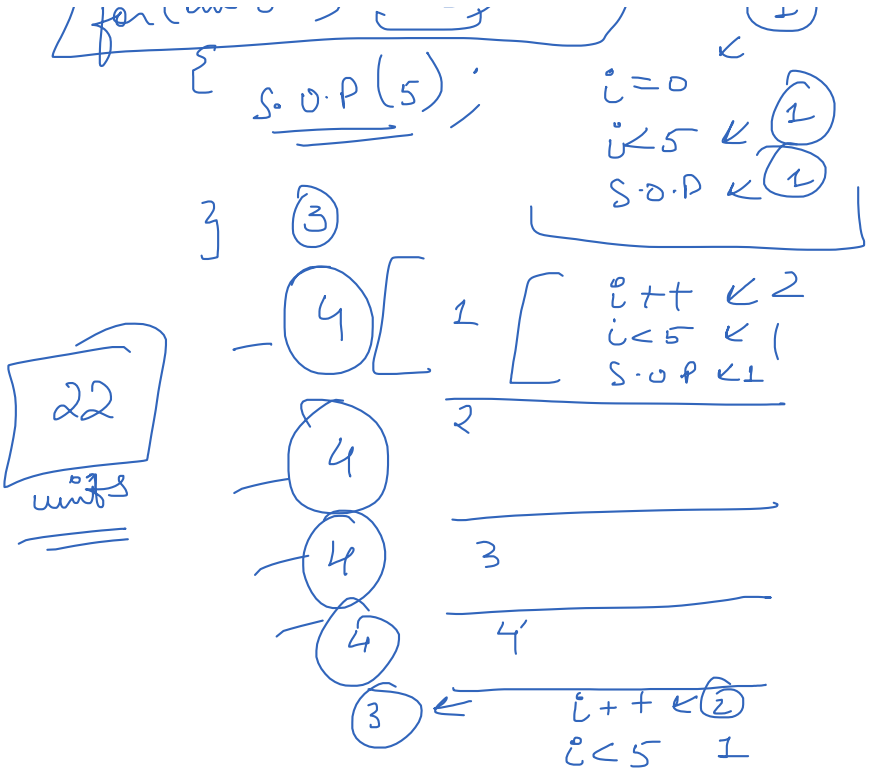
T.C ?

i = i+1

2. Loops

```
for(int i=0; i<5; i++)
{
  S.O.P(5);
}
```

i=0 → 1



3 Notations

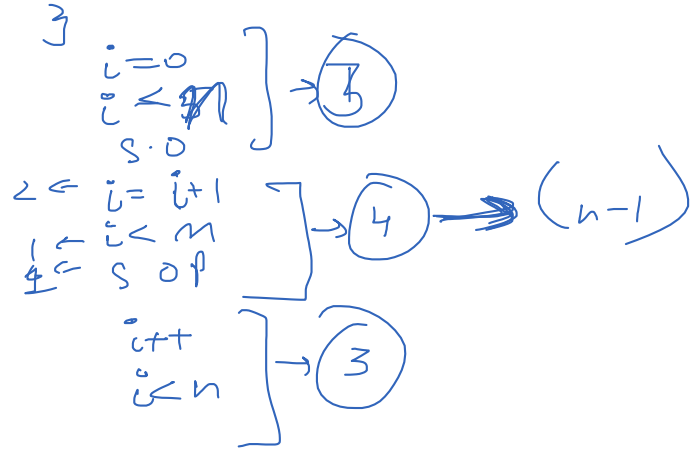
1 Big-Oh Notation

Worst Case Complexity

Constant time → O(1)

10 units → O(1)

```
for(int i=0; i<n; i++)
{
    S.O.P("Hello");
}
```



→ 3 + 4(n-1) + 3

⇒ 3 + 4n - 4 + 3 = 4n + 6 - 4

≈ 4n + 2

$3 + 4n - 4 + 3 = 4n + 6 - 4$

$= 4n + 2$

$4n+2 \Rightarrow O(4n+2)$

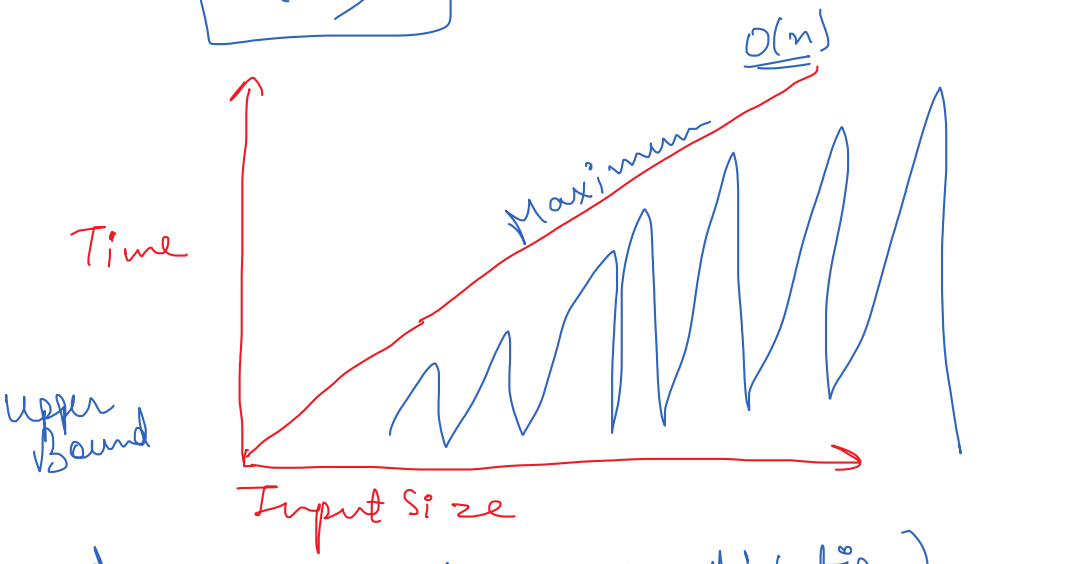
$O \rightarrow$ Worst Case

$n \to \infty$

$4n$ (+2)

$O(n)$

Code $\propto$ size of input

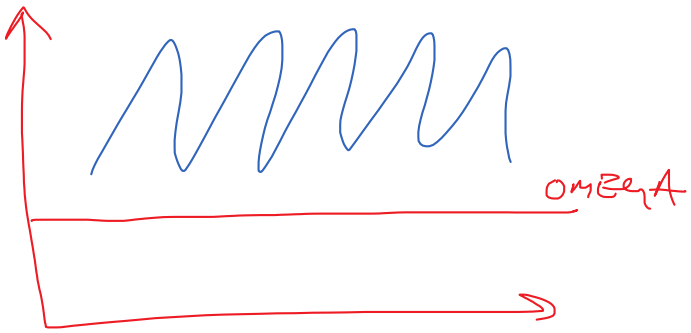


2nd Ω (OMEGA Notation)

(Lower Bound)

$f(n) \geq c \times g(n)$

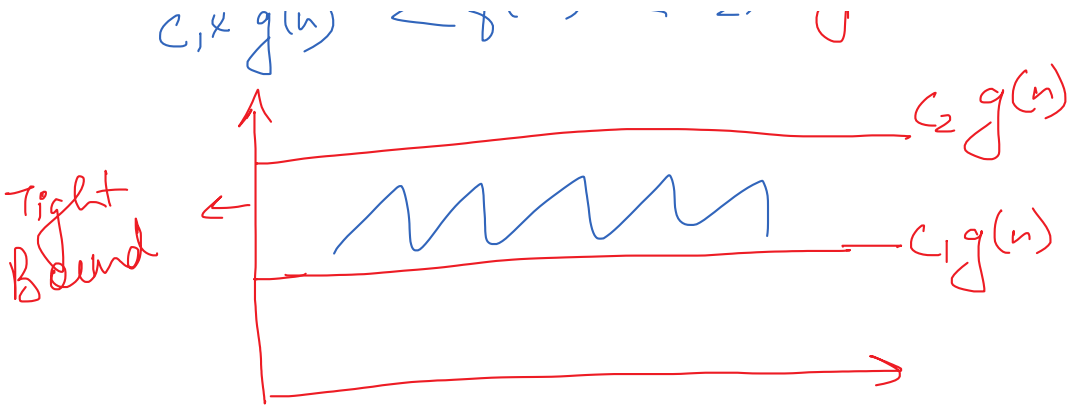
OMEGA Notation



3 Θ-Notation

$c_1 \times g(n) < f(n) < c_2 \times g(n)$

$g(n)$



Sorting → Time Complexity
→ Space Complexity → Inplace
given: 1 2 3 4 5
create: Extra space

1. Selection Sort

∞ ∞ ∞ ∞ ∞

2 4 7 9 10 — Extra

10 7 9 2 4

1 → Find Min index
2 → Swap with start pos

i=0 i=1 i=2 i=n-2

After 1 pass: 2 7 9 10 4 — n-1, min index

2 4 9 10 7 — $(n-2)^{th}$ index

2 4 7 10 9

2 4 7 9 10

$i <= n-2$

$i < n-1$

2. Bubble Sort

10 7 9 2 4

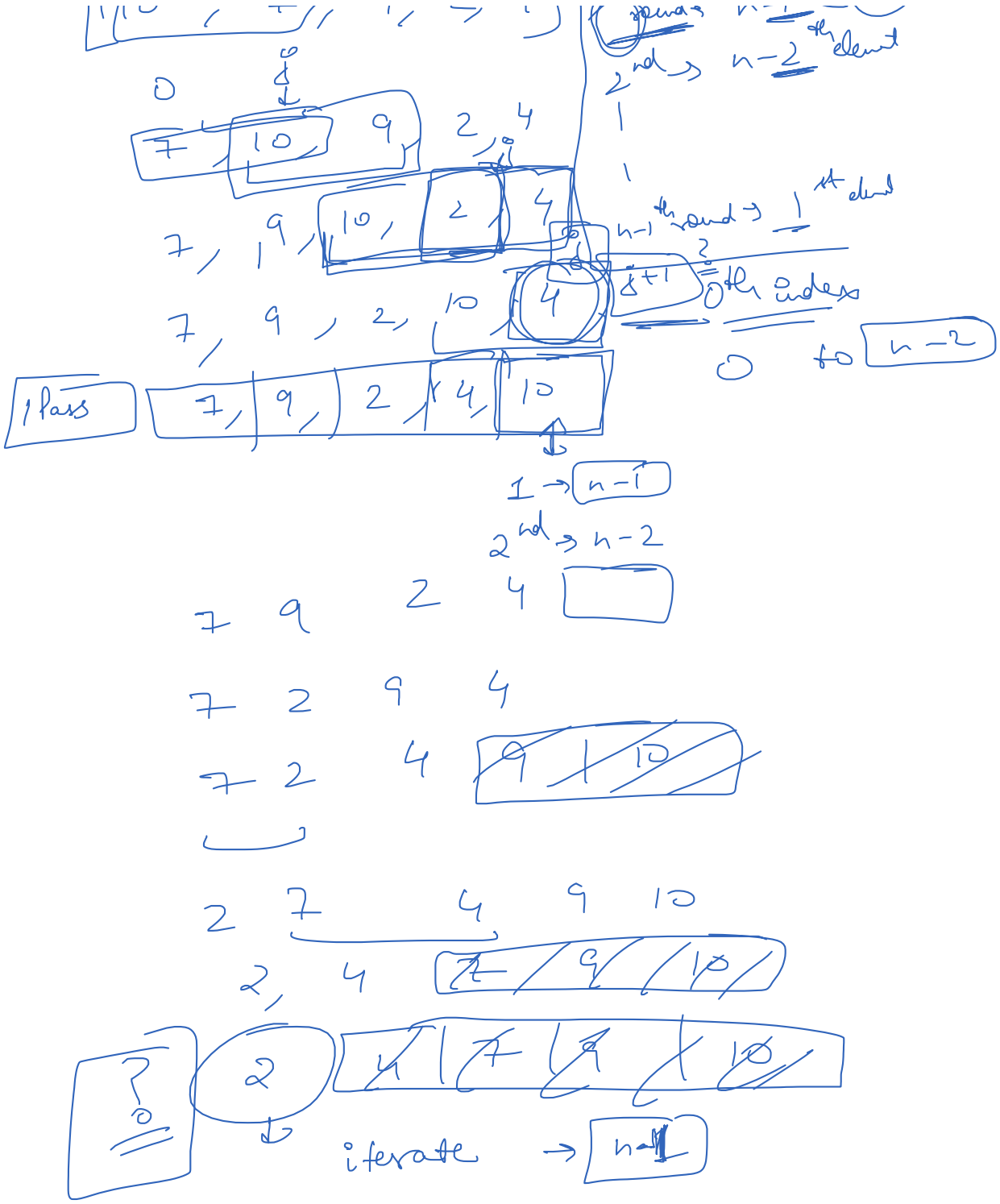
i

4 passes

last index: j

1st round: n-1 elements

2nd: n-2



Recursion

↳ A function calling itself directly / indirectly.

Direct	Indirect
void fun( )	void A( )

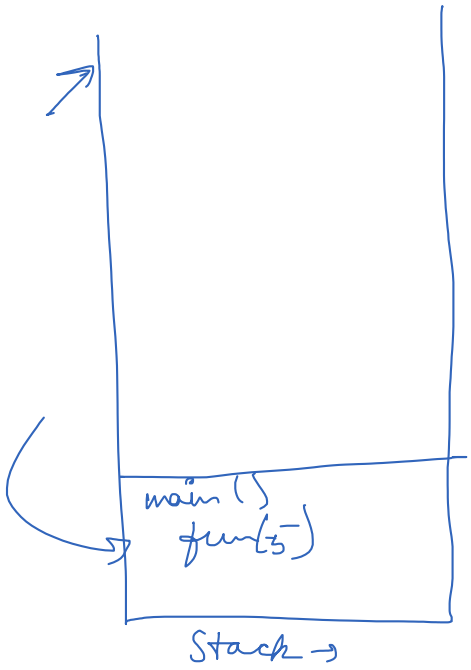
```
void fun(.)
{
  fun(.);
}
```
→ Base Case()

```
void A()
{
  B(); Base Case()
}
void B()
{
  A();
}
```

Base Case / Terminating Condition

Example

```
void fun(int n)
{
  S.O.P(n);
  fun(n-1);
}
```



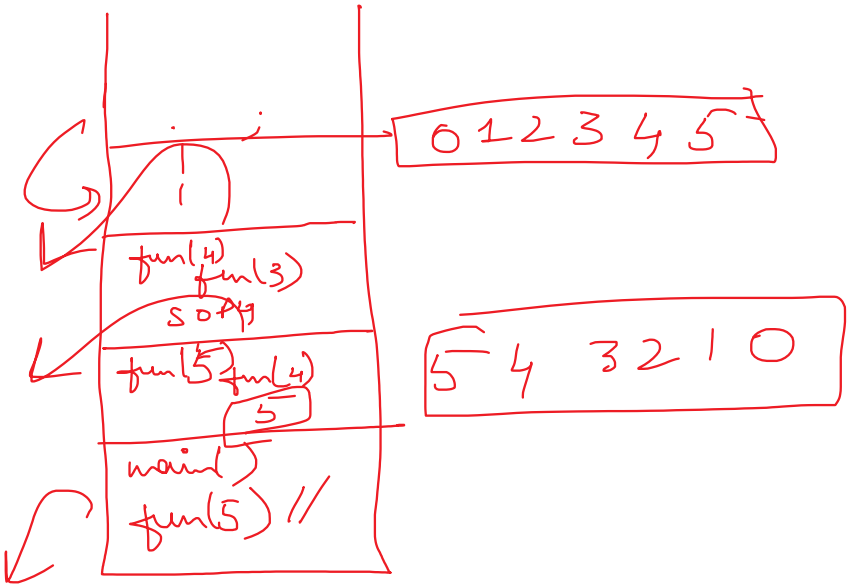
Print
5 4 3 2 1
0 -1 -2
-100 ∞
5 3 2 1 0

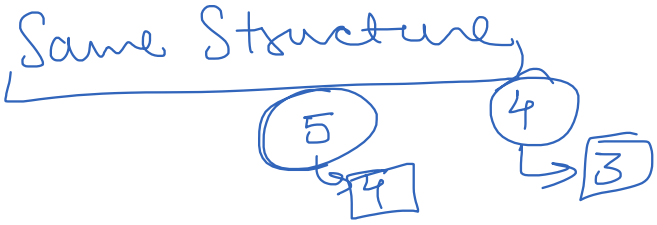
1 ⇒ Your Assignment
2 ⇒ Revise everything
3 ⇒ Types of Recursion *
4 ⇒ Print (0 to 5) in Assending order using recursion. And the first function call must be fun(5);

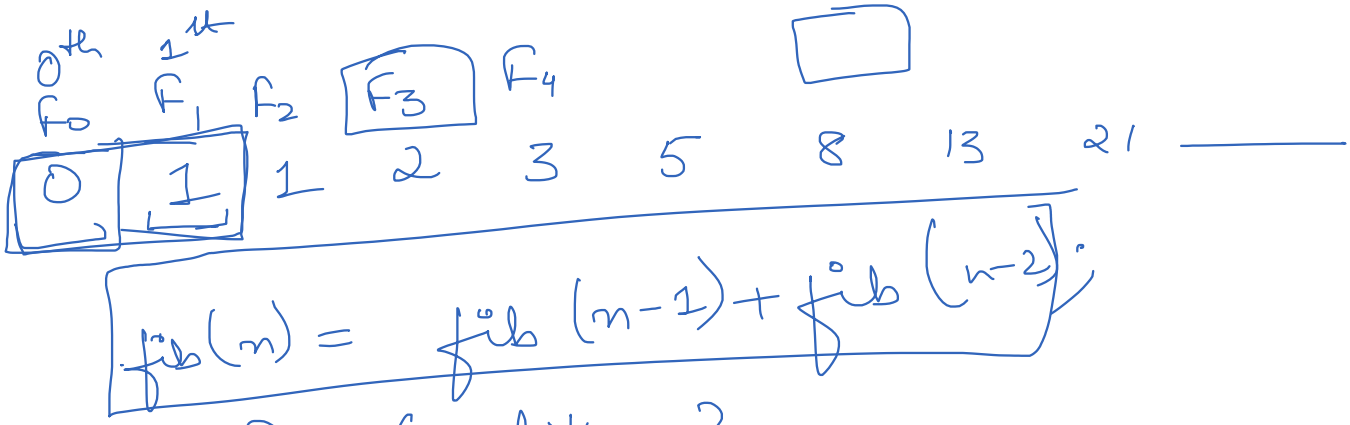
May - 22

22 May 2020 19:34

Ascending Print Recursion
Fibonacci
Binary Search(Iterative + Recursive)//Done
Sliding Window
Count pair sum
Doubts

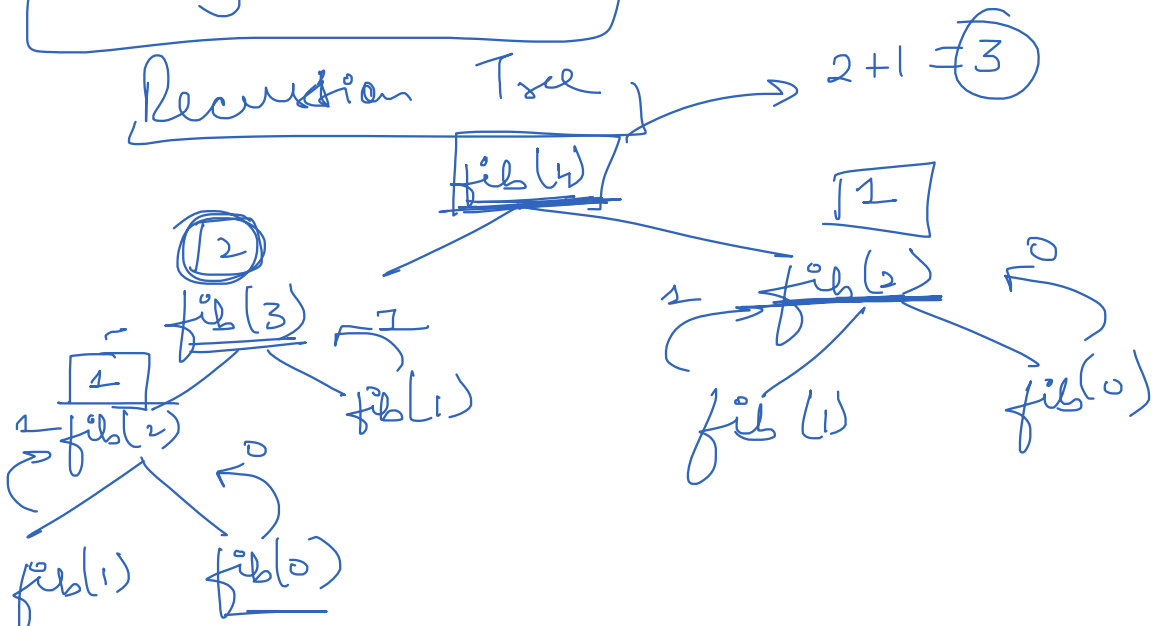






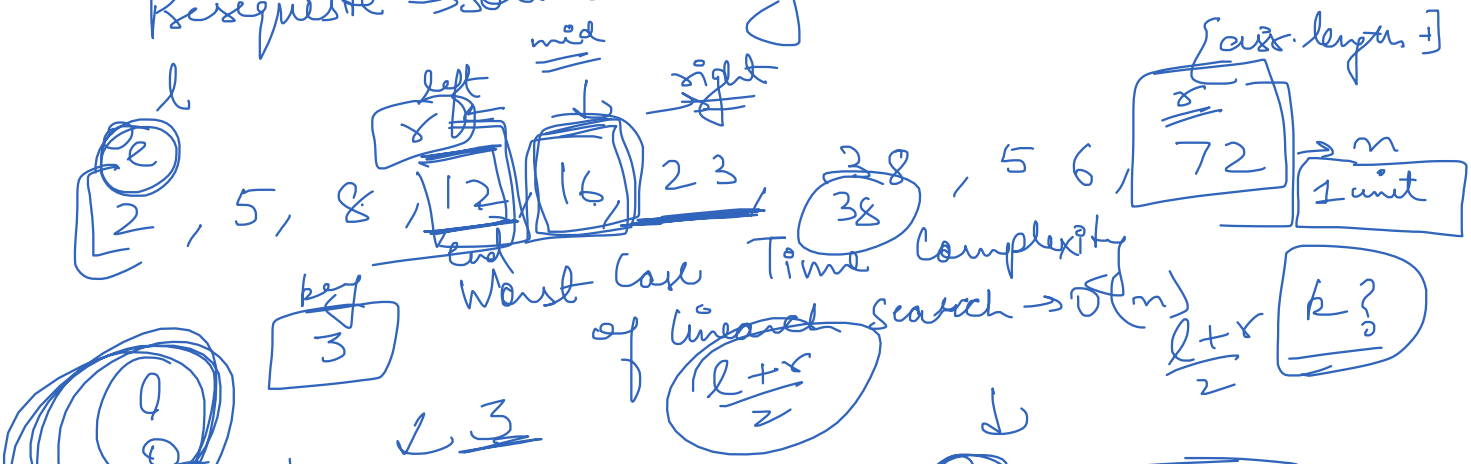
```
if (n<=1)
{ return n;
}
```

Recursion Tree



Binary Search

Prerequisite → Sorted Array



= No. of steps
= k

Let us suppose k steps

We stop when search space = 1

$O(k)$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log n = \log 2^k$$

$$\log n = k \log 2$$

$$k = \frac{\log n}{\log 2} \quad \left[\frac{\log a}{\log b} = \log_b a\right]$$

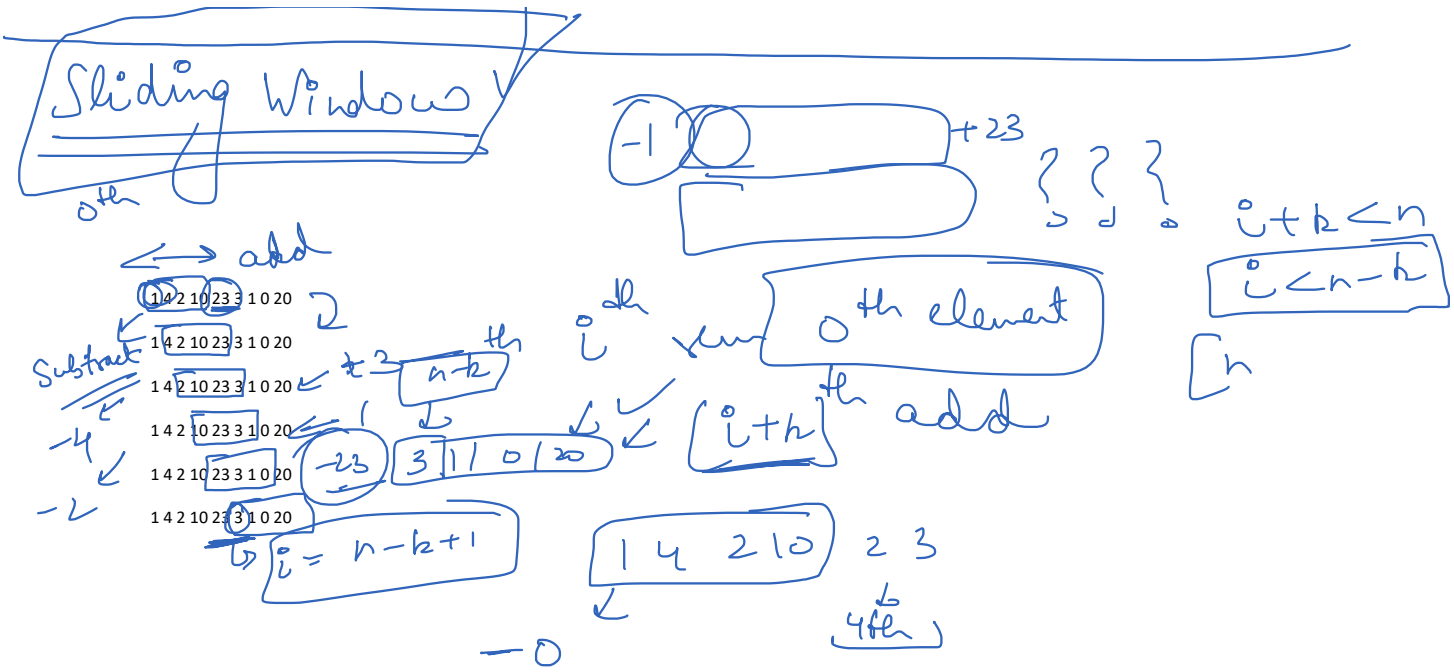
$$k = \log_2 n$$

$$T.C = O(\log_2 n)$$

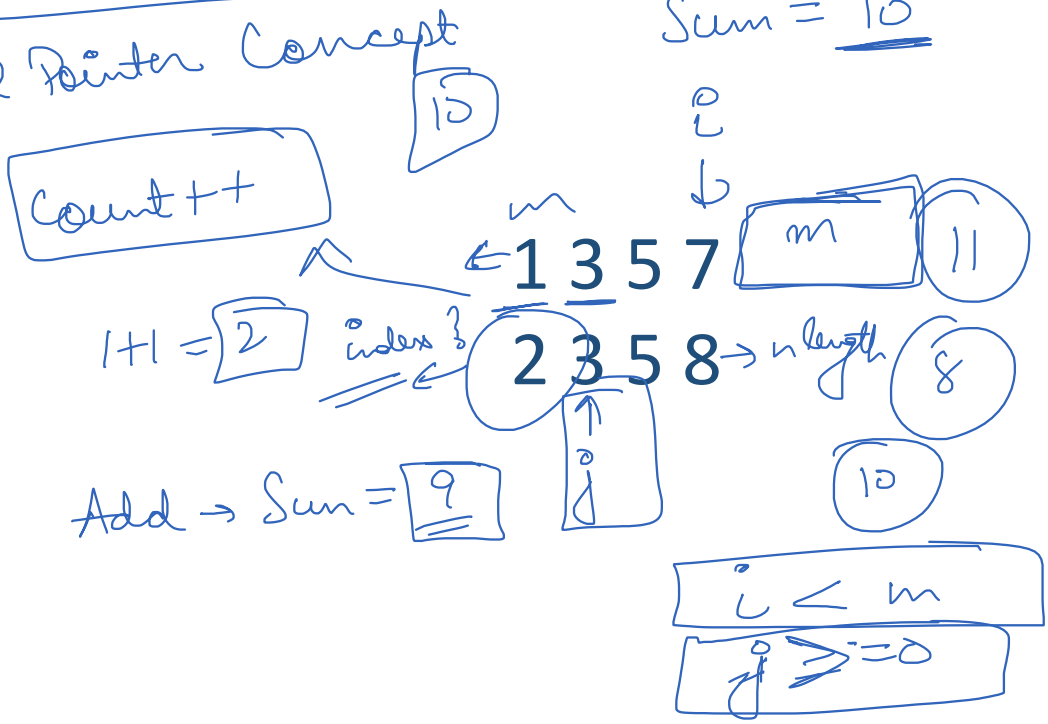
n	O(n)	O(log n)
1	1	1
10	10 ms	$\log_2 10 \rightarrow 3.32$ ms
100	100 ms	6.6 ms
1000		
10000		
10^6	10^6 sec = 1666 min	18 sec

Sliding Window

Sliding Window



2 Pointer Concept



Strings

25 May 2020 21:46

Reverse a String
Check Palindrome
Frequency Array
Anagrams
Check if two strings are rotations of each other
Confused pappu etc +1...

ARORA

Palindrome:
Same when written ulta

ARORA
MALAYALAM
GFG
121
1331
NAMAN
CIVIC

Frequency Array:

punitpundir

a-0	b-1	c-2	d-3			i-8		n-13		p-15	r-17	t-19	u-20					y-24	z-25					
0	0	0	1	0	0	2	0	2	0	2	1	1	2	0	0	0	0	0	0					

abab
baba

Char c='a'

aabb
aaaa

Size - 26

Int x =(int)c

aabb
aaaabb

Int x = (int)c-97

Character
a - 97 - 0
b-98 - 1
c-99-2

a-6
b-4

A - ++
B - --

A-65
a-97

If all types of characters including spaces and special characters are allowed:
Take array size as 256