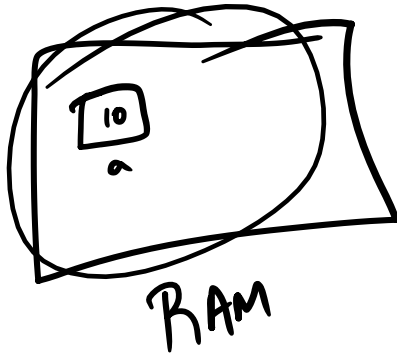


Space Complexity

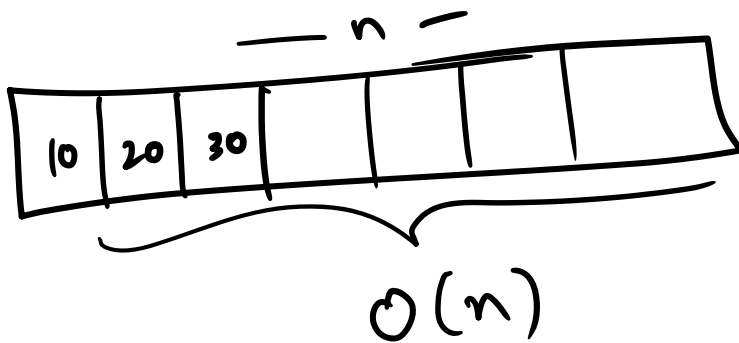
$a = 10$ — ① unit



$a = 10$ — 1
 $b = 20$ — 1
 $c = 30$ — 1

③

$\rightarrow O(1)$
 $c = 'x'$ — ①
 $c = \underline{\text{true}}$

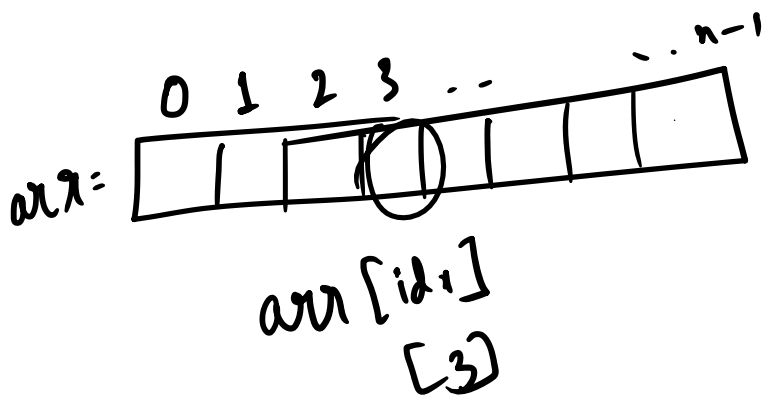
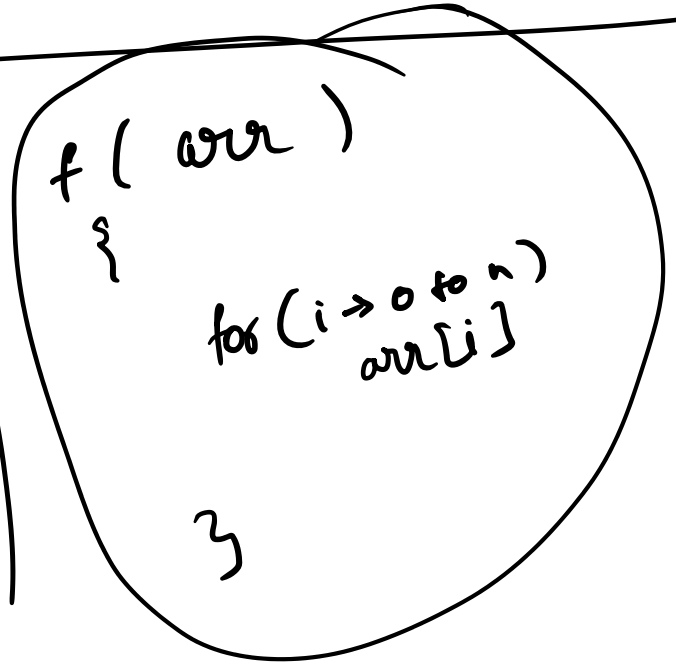
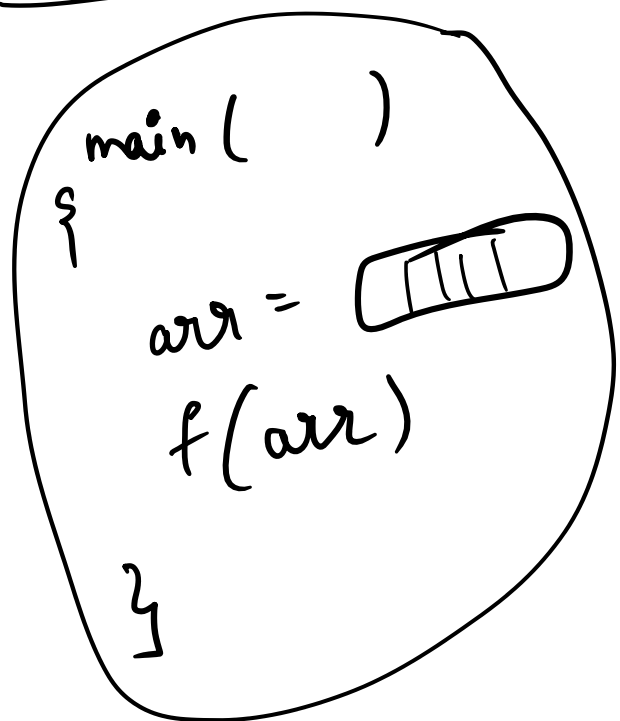


"hello - -"
 \Downarrow
 $O(n)$

for()
{
 for()
 {
 arr[i]
 [j]
 }
}

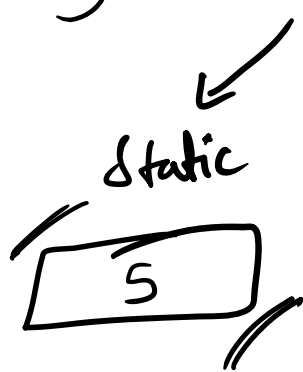
$a = 10$ —
 $a =$ []

3



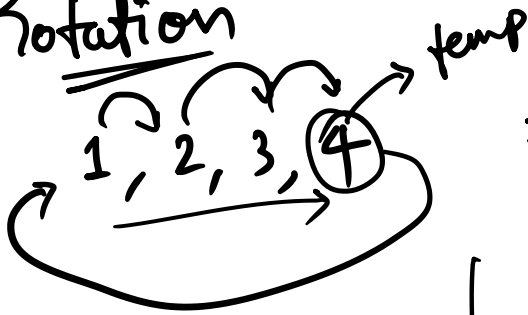
`arr[100]`
→ undefined

Pranklist
Shopping
Cart
Chat



JS
Dynamic
[]

Rotation



\Rightarrow

4, 1, 2, 3



TC \Rightarrow 2

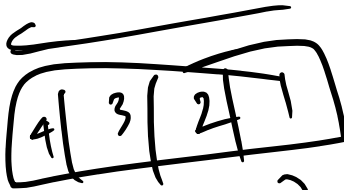
SC \Rightarrow 1

$arr[i] = arr[i-1]$

$arr[0] = temp$

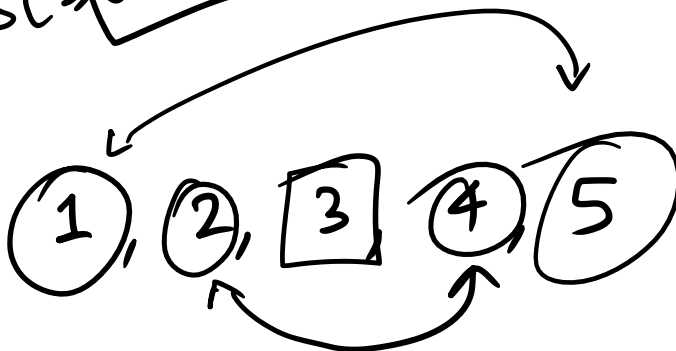
Reverse

1, 2, 3, 4 \leftrightarrow 4, 3, 2, 1



TC $\Rightarrow O(n)$

SC $\Rightarrow O(n)$



$$\left. \begin{matrix} \text{---}^n \\ \text{---}^n \end{matrix} \right\} 2n \Rightarrow O(n)$$

sort

3, 9, 1, 2 \rightarrow 1, 2, 3, 9

arr.sort() $\Rightarrow O(\underline{n \log n})$

["hello", "bottle", "apple"] \Downarrow ["apple", "bottle", "hello"]

Lexicographical Order

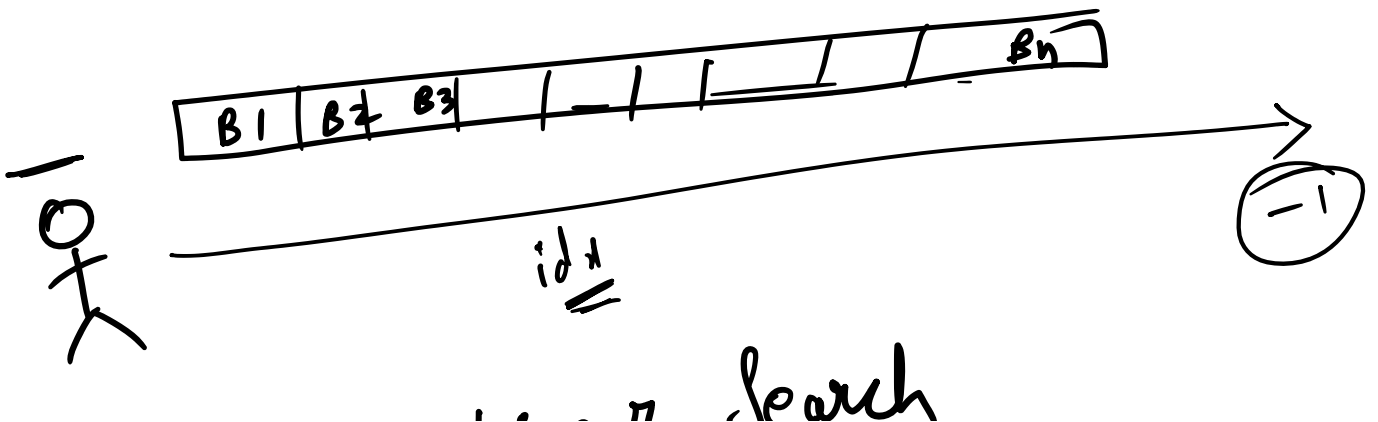
$x = 4$

[2, 5, 8, 9, 3, 8, 7, 6]

↓

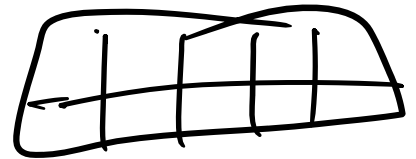
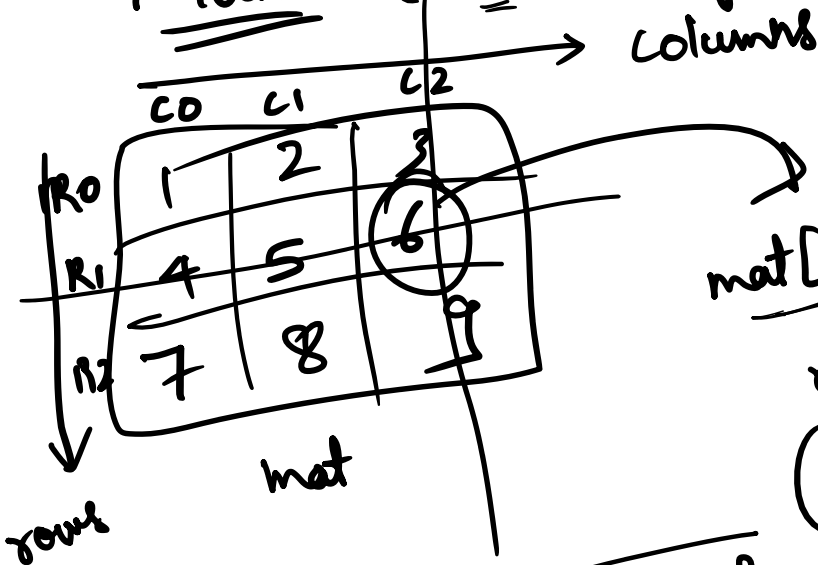
2

(-1)

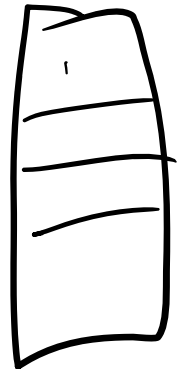


Linear Search

Matrix (2D-array)



2x5



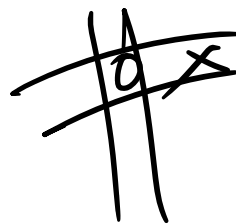
4x1

dimension \rightarrow rows \times col
(3) \times (3)

Square $\rightarrow r == c$

Rectangle $\rightarrow r \neq c$

Board Games \rightarrow Chess, Tic tac toe

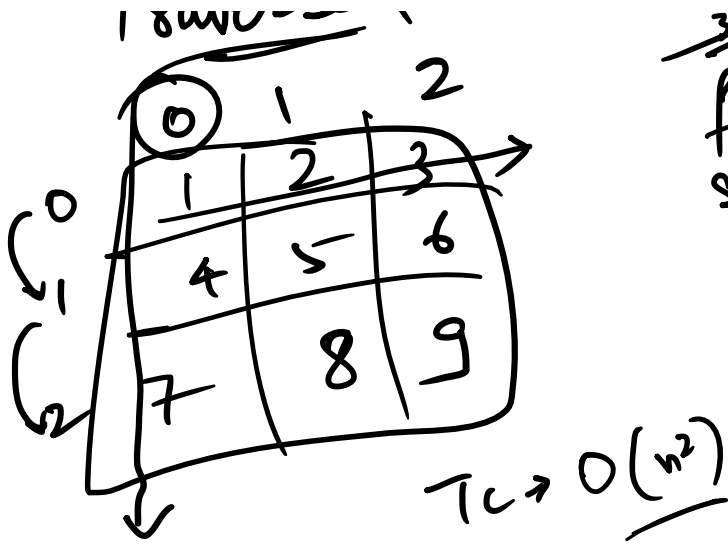


Traversal



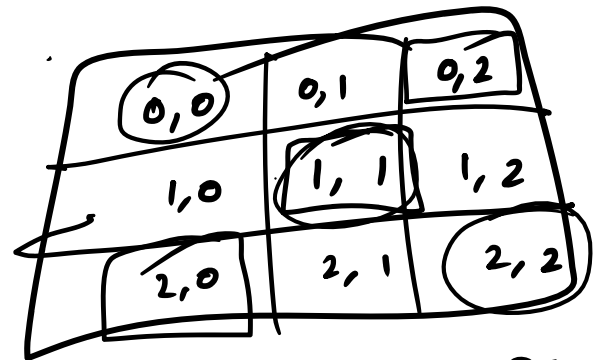
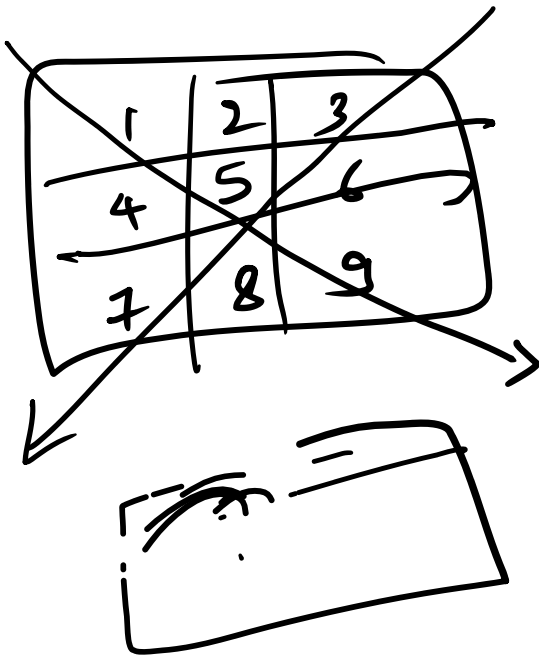
1 4 7
2 5 8
3 6 9

P (1: - n to n)



for ($i \rightarrow 0$ to n)
 {
 for ($j \rightarrow 0$ to n)
 {
 mat[i][j]
 }
 }
 }
 }
 }
 }

if ($i == j$)
 mat[i][i]



for ($i \rightarrow 0$ to n) $\rightarrow O(n)$
 mat[i][i] $\rightarrow j = i$

$$i + j = n - 1 - i$$

$$j = n - 1 - i$$

mat[i][n-1-i]

15 \rightarrow target

