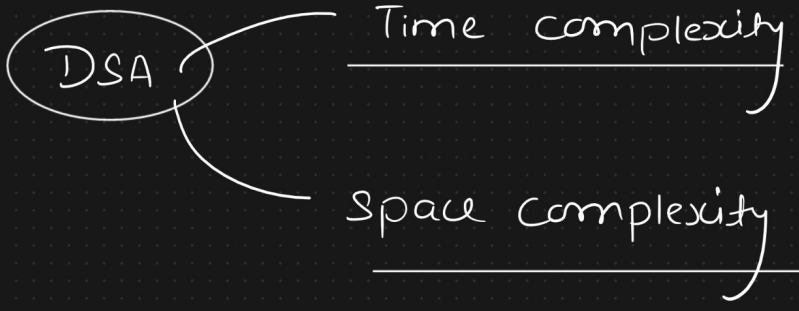
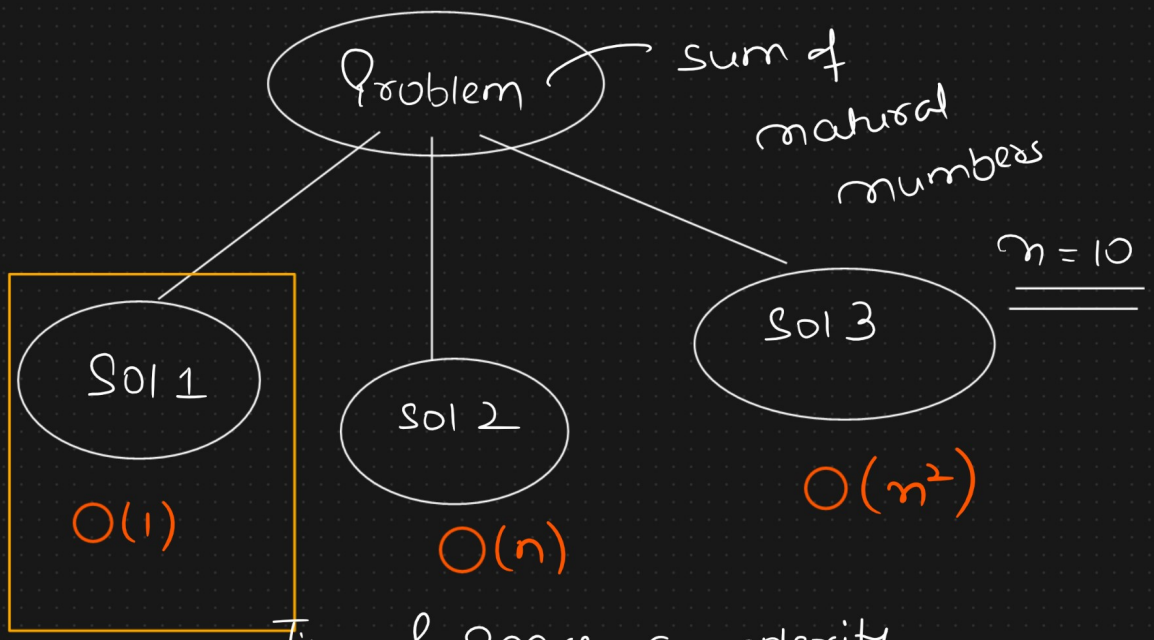
Week-1 DSA





Time & Space complexity

↳ very less

1) substitution
2) Recurrence Tree
3) Master's Theorem

Recurrence Relation

Sum of natural numbers

$n = 10$

Approach 1

↳ O(n)

```
int sum = 0;
for (i = 0 to 10){
    sum += i;
}
return sum;
```

Approach 2

↳ O(1)

$\frac{n(n+1)}{2}$ = sum

return sum;

3) Reversal

4) Duplicate

Array

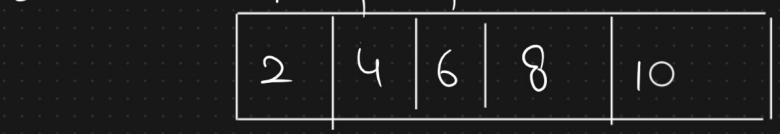
- 1D Array
  - 1) Palindrome
  - 2) Missing no.
- 2D Array (rows & columns)

Linear Data Structure

Matrix

↳ Contiguous Storage

Base add 1000



↳ Rotation of Matrix

↳ Prefix sum
- 1D Array
- 2D Array
  - Brute force Approach
  - Optimized approach

Binary Search

$O(\log_2 n)$ — Logarithmic time complexity

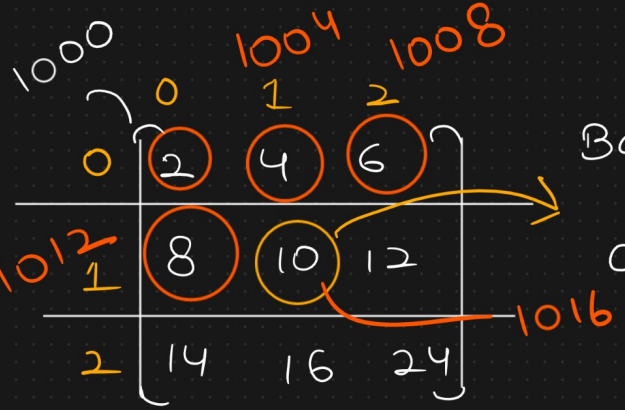
$nR = 3$

$nC = 3$

$LB_0 = 0$

$LB_c = 0$

Row major form



Base add = 1000

add(i, j)

add(1, 1) =

Memory Addressing

$$1000 + \left((1-0) * \overset{nC}{3} + (1-0) \right) * \underset{4}{1}$$

$$1000 + (4 * 4) = 1016$$

Priority 1 — Monday — videos

↳ Implementation (on your own)

Priority 2 — Assignment — Solutions

Priority 3 — Leetcode — Array ↳ Easy

GP Series

$O(4n) = O(n)$

→ Common ratio

$r > 1$

$r < 1$

→ Sum of GP series

$(1 + mc - c)$ $O(m)$

90% clockwise

Input matrix

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Output

7	4	1
8	5	2
9	6	3

Step1 – Transpose

l_i r_i

$\begin{cases} l_i = 0 \\ l_i = 2 \end{cases}$

	0	1	2
0	1	4	7
1	2	5	8
2	3	6	9

3×3

	0	1	2
0	7	4	1
1	8	5	2
2	9	6	3

$i = 0$ $i = 1$ $i = 2$

swap(arr(i)(li), arr(i)(ri))

$\begin{cases} l_i = 1 \\ l_i = 1 \end{cases}$

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

→ 2×4

$$\begin{bmatrix} 5 & 1 \\ 6 & 2 \\ 7 & 3 \\ 8 & 4 \end{bmatrix}_{4\times 2}$$

final output

	li	ri
0	5	1
1	6	2
2	7	3
3	8	4

final result

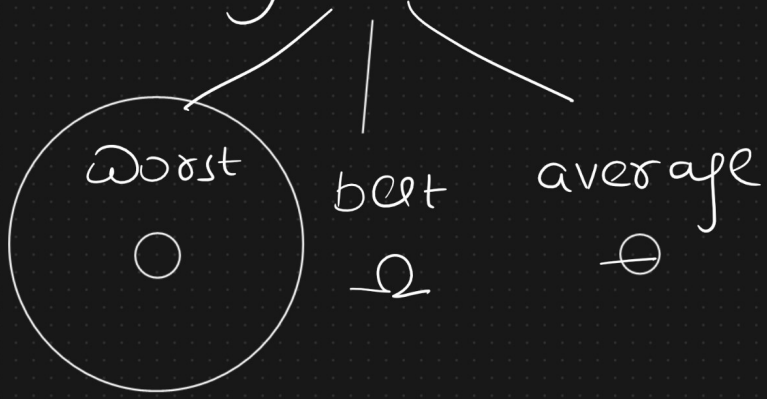
$$\begin{cases} l_i = 1 \\ r_i = 0 \end{cases}$$

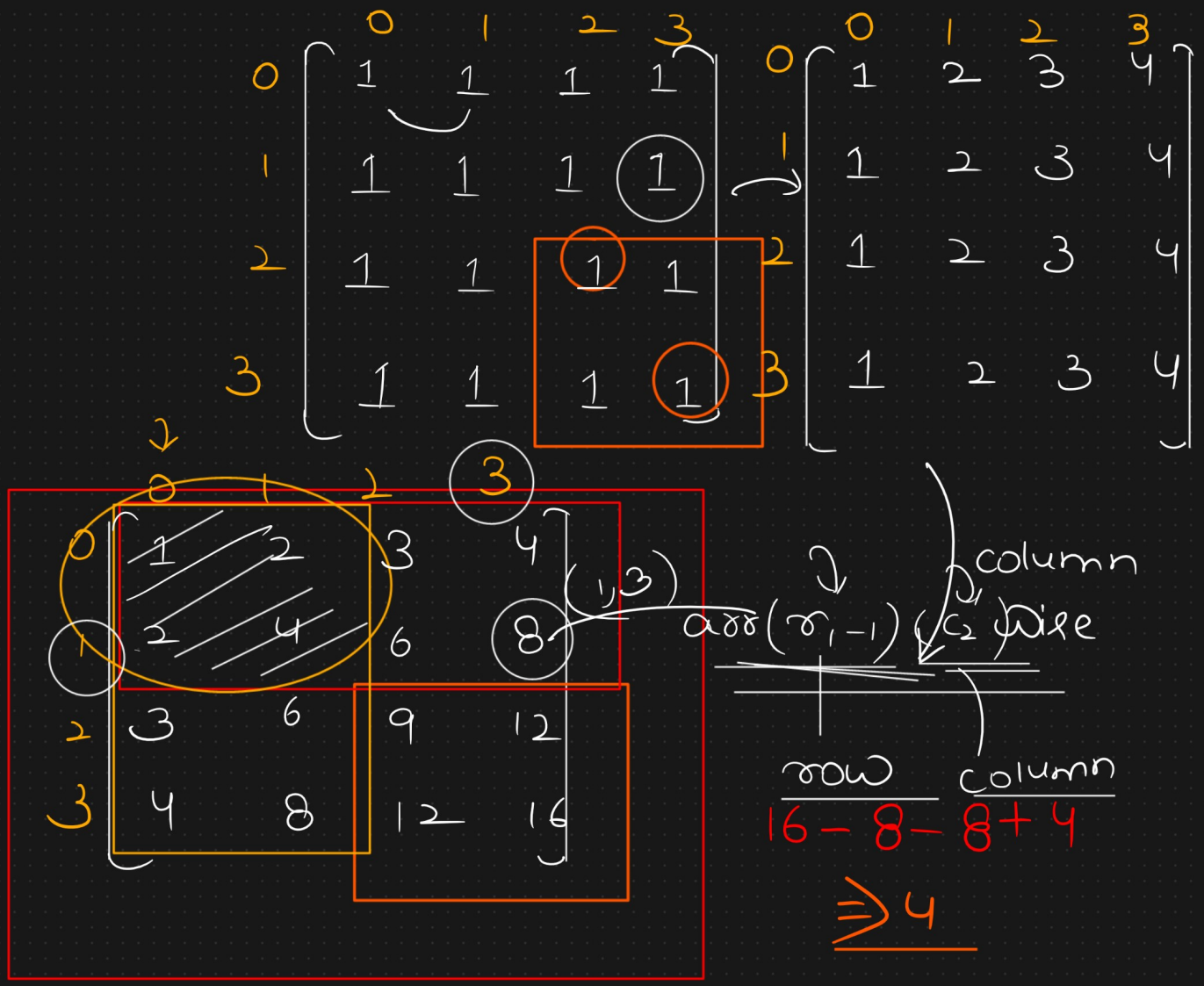
```
while(li < ri)
{
    swap(arr(i)(li), arr(i)(ri))
}
li = li+1
ri = ri-1
```

$$\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{Bmatrix} \xrightarrow{\text{transpose}} \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$ transpose

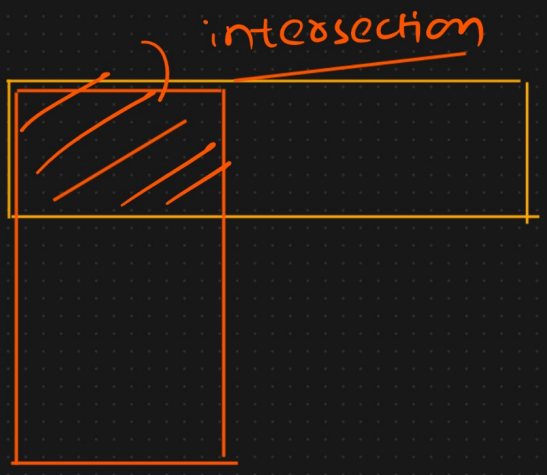
Asymptotic Notations





$r_1 = 2$ $c_1 = 2$

$r_2 = 3$ $c_2 = 3$



original array

2	1	4	8	10

Algorithm / Logic

Sorting in increasing order

new array

1	2	4	8	10

display

Space complexity = O(n)