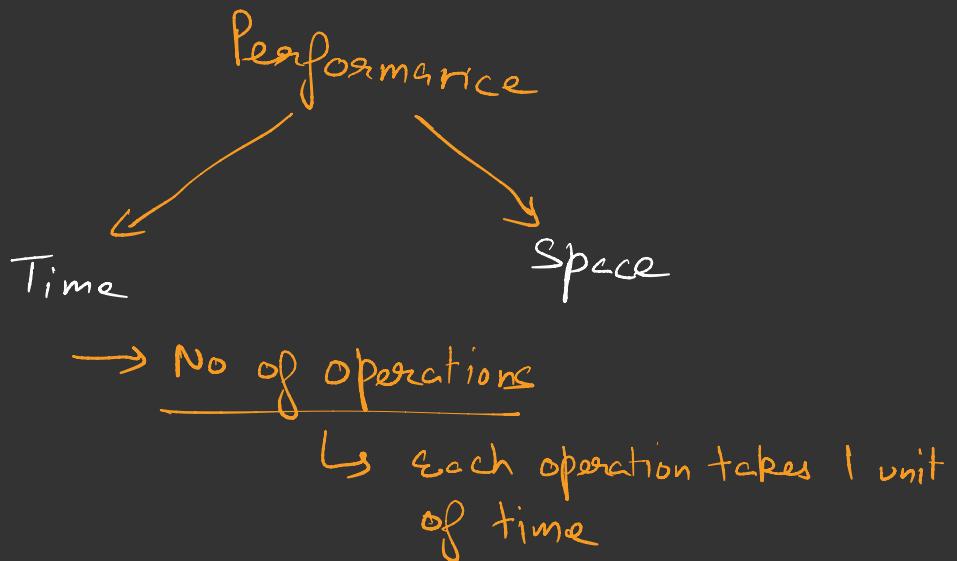


Time & Space Complexity



`Sys("Hello World")` → 1 operation

Sys0(1)
 Sys0(2)
 Sys0(3)
 Sys0(4)
 Sys0(5)

} → 5 operations
 ↗ 5 unit of time

```
for (i=1; i<100000; i++) {
```

\S sys(i) — 1 operation $\rightarrow 10^5$ times
 10^5 operations

Size of input(n) & no of operations

$n = 10$ linear 10

$n = 10$ Quadratic (n^2) $\Rightarrow 100'$

$n = 10$ Cubic (n^3) $\Rightarrow 1000$

Linear Search

```
for (int i=0; i < arr.length; i++) {  
    if (arr[i] == val) { // 1 operation  
        return i;  
    }  
}
```

$n = 5$

0	1	2	3	4
1	2	3	4	5

Best case $\Omega(1)$

no of operations = 1

Average case $\Theta(n)$

$$\frac{1+2+3+4+5+\dots+n}{n}$$

Worst case $O(n)$

$$\text{avg} = \frac{n(n+1)}{2n} \Rightarrow \frac{n+1}{2}$$

\downarrow
max time / max no

n compression

of operations your algo will perform

```
bsv main() {
```

```

Scanner s = new Scanner();
int n = s.nextInt();
for(int i = 1; i <= n; i++) {
    System.out.println(i);
}

```

$n+2$
 Operations
 $O(n)$

[] = **kopatizie**
[] **unit x n**
[] **n unit**
[] **no operations**

$$\textcircled{1} \quad f(n) = n + \textcircled{k} \Rightarrow \text{constant} \quad / / \quad O(n)$$

$$\textcircled{2} \quad f(n) = 4n + 3 \Rightarrow O(4n) \Rightarrow O(n)$$

$$\textcircled{3} \quad f(n) = 8n^2 + 6n + 3 \Rightarrow O(8n^2) \quad O(n^2)$$

11 Remove non significant powers
and constants

② Nested loop

```
for(i=1; i<=n; i++) {
```

```
for(j=1; j<=n; j++) {  
    SysOut(n);
```

۳

1

1

1

285 ... 5

2

XXIII - 4

No. of Operations

5

h

$n \quad i \geq 8, \dots, n \quad n$

$$\begin{aligned} \text{total} &= \underbrace{n + n + n + \dots + n}_{n \text{ times}}, \\ &= n(1 + 1 + 1 + \dots + 1) \\ &= n \times n \Rightarrow n^2 \end{aligned}$$

Time $\Rightarrow O(n^2)$

③ Nested n, m

$\text{for } (i=1; i \leq n; i++) \{$

$\text{for } (j=1; j \leq m; j++) \{$

$\} \quad \text{System}(H_i);$

$\}$

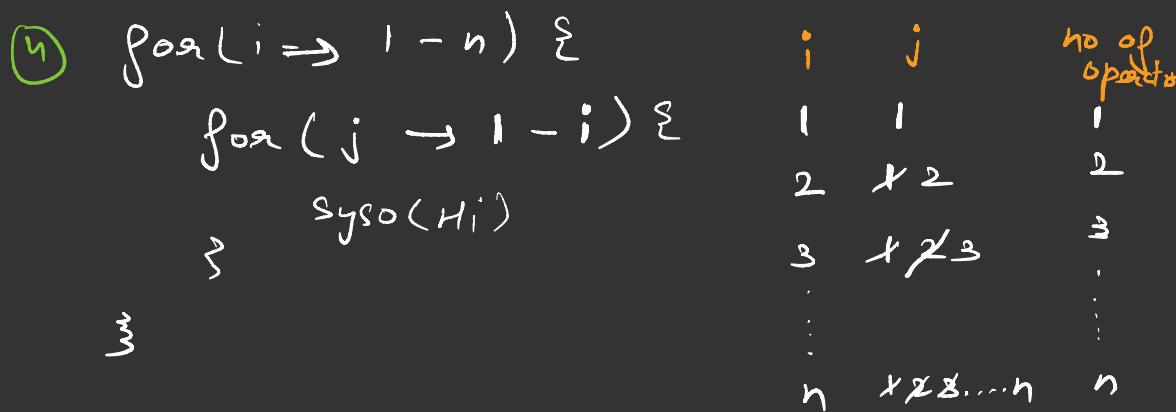
i	j
1	1 ... m
2	1 ... m
3	1 ... m
...	...
n	1 ... m

$\underbrace{m + m + m + \dots + m}_{n \text{ times}}$

$n \times m$
 $\Rightarrow O(nm)$

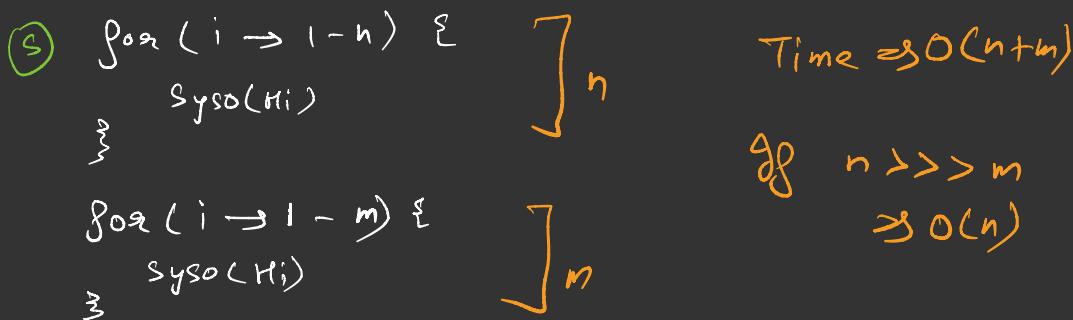
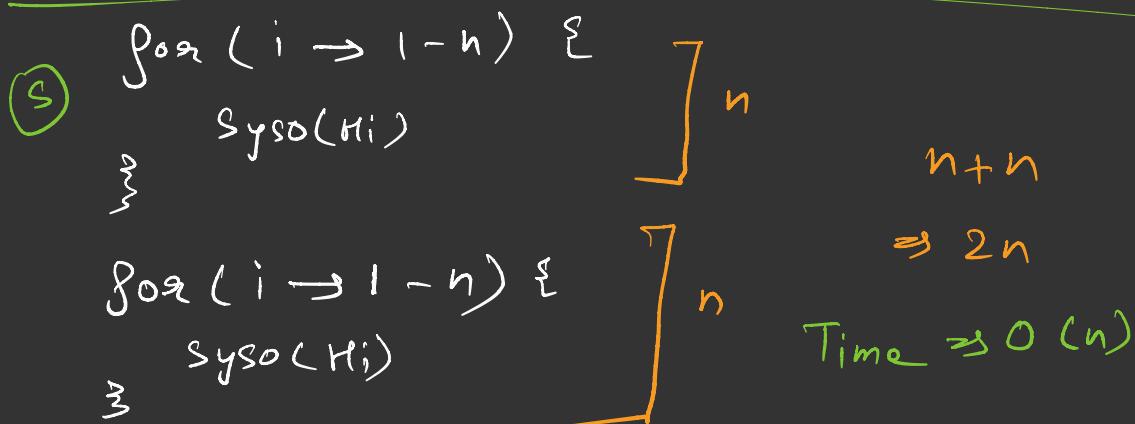
$n \gg m$

$\Rightarrow O(nm)$



$$1 + 2 + 3 + 4 + \dots + n \Rightarrow n \times \frac{(n+1)}{2} = \frac{n^2+n+1}{2}$$

$$\Rightarrow O(n^2)$$



$\text{for } i = 1 \text{ to } n \{$

$\text{fun2}()$

}

$\text{fun2} \rightarrow \{$

:

$\text{fun1}();$

}

] \rightarrow operation

$\text{fun2} \Rightarrow O(n)$

$\text{fun3} \rightarrow \{$

$\text{for}(i = 1 \text{ to } n) \{$

} $\text{fun}(n)$

}

n times
n operation

$\Rightarrow O(n \times n)$
 $O(n^2)$

;

1

2

3

:

n

n
 n
 n
 n
 n

$n + n + n + n \dots - n$
n times

n^2

n	n^2	n^3	n^4
1	1	1	1
2	4	8	16
3	9	27	81
10 ⁵	10 ¹⁰	10 ¹⁵	10 ²⁰

$$O(1) < O(\log n) < O(n)$$

$$< O(n \log n) < O(n^2)$$

$$< O(n!) < O(2^n)$$

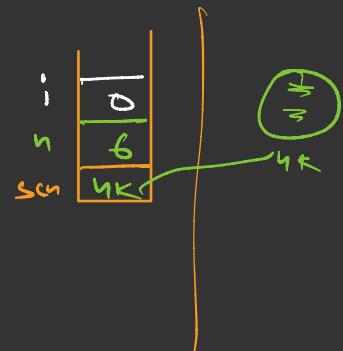
Space Complexity

Scanner scn = new Scanner();

int n = scn.nextInt();

for (int i = 0; i < n; i++) {

 System.out.println(i);
}



Space

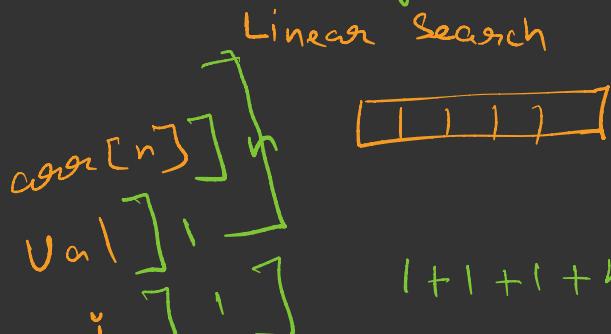
Space Complexity

Auxiliary Space

→ Space taken by
your program to execute

Space taken by
your algo apart
from input

→ Input + Auxiliary Space



linearS(int arr, int val)

{

}

$$1 + 1 + 1 + n \\ \Rightarrow 3 + n \\ O(n)$$