

TIME COMPLEXITY

Time & Space Complexity



Time



This Space

memory

Time Complexity

21.28

Relation between Input Size & Running Time (Operations)



Time Complexity

Relation between Input Size & Running Time (Operations)

↓
 n

m

Scanner → input a variable "n"

```
for (int i = 0 to n)
```

```
{
```

```
    print("hello"); → ①
```

```
}
```

} 1x n times

```
main() {
```

```
    print("hello");
```

```
}
```

```
main()
```

```
{
```

linear

quadratic

cubic

log

89.8

1 Time Complexity

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt(); }  
}
```

```
for(int i=0; i<n; i++) {  
    System.out.println("hello");  
}  
}
```

input n

n

time

→ n times



1 Time Complexity

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt(); }  
  
for(int i = 1; i <= n; i++) {  
    System.out.println("hello");  
}
```

→ n times

input n

n

time complexity \propto input n

2 Time Complexity

BEST CASE

AVERAGE CASE

WORST CASE

Numbers : {1, 2, 3, 4, 5}

search for '1'



2 Time Complexity

BEST CASE

AVERAGE CASE

WORST CASE

Numbers : {1, 2, 3, 4, 5}

search for '1'

(i) 0 1 2 3 4 5

↓
pos = 1

(1) { 1 operation
1 unit of time

(ii) 0 1 2 3 4 5 { 1+2+3+4+5
2 3 4 5 1 2 3 4 5 }

$$\begin{aligned} & \frac{[1+2+\dots+n]}{n} \\ & \Rightarrow \frac{n(n+1)}{2} \approx n \cdot \left(\frac{n+1}{2} \right) \end{aligned}$$

2 Time Complexity

BEST CASE $\rightarrow \Omega(1)$

AVERAGE CASE $\rightarrow \Theta\left(\frac{n+1}{2}\right)$

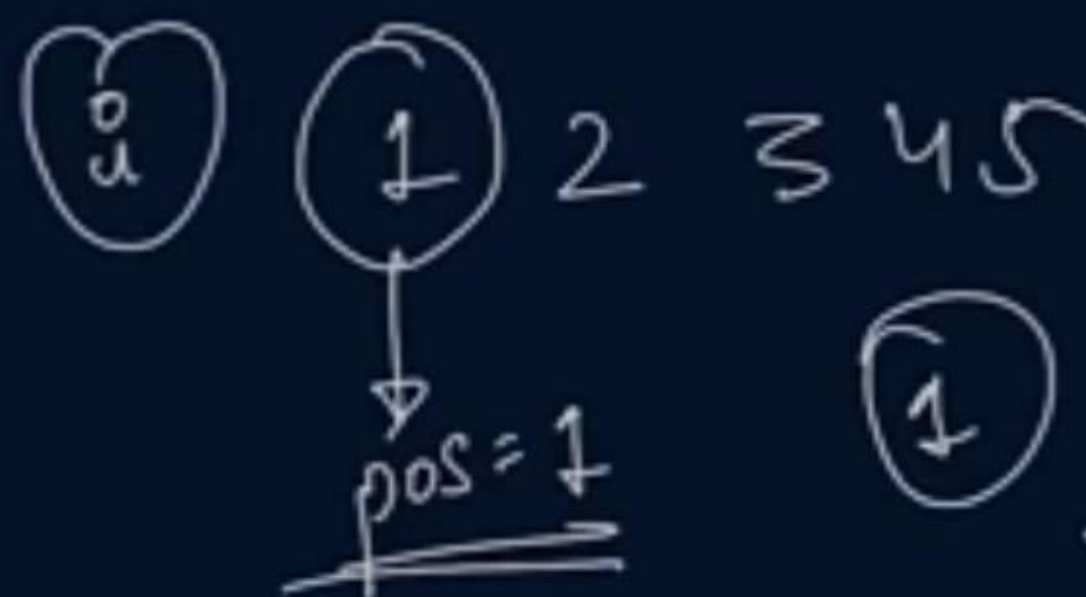
WORST CASE $\rightarrow O(n)$

(iii) worst case $n=5$
 $[5, 4, 3, 2, 1]$
 $\propto n$

10^5

Numbers : $\{1, 2, 3, 4, 5\}$

search for '1'



$\textcircled{1} \left\{ \begin{array}{l} 1 \text{ operation} \\ 1 \text{ unit of time} \end{array} \right.$

(ii) $\begin{matrix} 2 & 1 & 3 & 4 & 5 & \{ 1 & +4+5 \\ 2 & 3 & 4 & 2 & 5 & \vdots & \end{matrix}$

3 Time Complexity

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt(); }  
  
    for(int i=0; i<n; i++) {  
        for(int j=0; j<n; j++) {  
            System.out.println("hello");  
        }  
    }  
}
```



dynamic

7 Time Complexity

Compare :

$O(n)$

$O(n^2)$

$O(n^3)$

$n=1$

1

1

1

$n=2$

2

4

8

$n=3$

3

9

27

↓
 10^5

10^5

10^2

10^{30}

1 Space Complexity

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt();  
  
    for(int i=0; i<n; i++) {  
        System.out.println("hello");  
    }  
}
```

