Time complexity 2

Time & Space Complexity
Asymptotic Analysis
Big OH
Time Limit Exceeded

6

AGENDA

CRUX OF TIME COMPLEXITY : # iterations (No. of iterations)

$N \xrightarrow{/2} 1$, $\underbrace{\quad}_{\log_2 N}$

Maths Concepts

QUIZ 1: $\log_2 N$ → No. of times we need to divide N by 2 to reduce it to 1.

[a b] ⇒ range inclusive of a & b

(a b) ⇒ range exclusive of a & b

[a b) ⇒ a is included, b not included

QUIZ 2 [3 10]

$\underbrace{3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10}_{8 \text{ nos.}}$

[a b] → $b-a+1$

[1 7] → $7-1+1 = 7$

[12 23] → $23-12+1 = 12$

Arithmetic Progression

series where adjacent term has same difference

$\underbrace{4, \quad 7}_{3}, \underbrace{\quad 10}_{3}, \underbrace{\quad 13}_{3}, \underbrace{\quad 16}_{3}$

$\Rightarrow \underbrace{a, \quad a+d}_{d}, \underbrace{\quad a+2d}_{d}, \underbrace{\quad a+3d}_{d}, \ldots$

$a \rightarrow$ first term
$d \rightarrow$ common difference

N terms

#Todo.

Sum of N terms of AP

$$\frac{N}{2}\left[2a + (n-1)d\right]$$

Geometric Progression

series where each adjacent term has same common ratio

$\underbrace{5, \quad 10}_{2}, \underbrace{\quad 20}_{2}, \underbrace{\quad 40}_{2}, \underbrace{\quad 80}_{2}, \underbrace{\quad 160}_{2} \ldots$

$\underbrace{a, \quad ar}_{r}, \underbrace{\quad ar^2}_{r}, \underbrace{\quad ar^3}_{r}, \ldots$

$a \rightarrow$ first term
$r \rightarrow$ common ratio

Sum of GP

N terms $\dfrac{a(r^N-1)}{r-1}$ $\quad r>1$

$\dfrac{a(1-r^N)}{1-r}$ $\quad r<1$

log Basics

$\log_a x \Rightarrow$ No. of times we need to divide x by a till $x \to 1$

log basics → link

$$\log_a a^x = x$$

Que.

```
int func(int N){
    S=0
    for(i=1; i<=N; i++){
        S=S+i
    }
    return S
}
```

$i : [1 \quad N]$

#iterations:

$N-1+1 = N$

$O(N)$

Que.

```
void func(N, M){
    for(i=1; i<=N; i++){
        if(i%2==0){
            print(i)
        }
    }
    for(j=1; j<=M; j++){
        if(j%2==0){
            print(j)
        }
    }
}
```

$i : [1 \quad N]$

#iterations: N

$j : [1 \quad M]$

#iterations : M

#iterations : $M+N+M^2$

$O(M+N)$

$O(M^2+N)$

$O(M+ \cdot$

Que. $N>0$

```
int func(N) {
    S = 0
    for(i=1; i<=N; i=i+2){
        S = S+i
    }
    return S
}
```

i

1
3
5
7
⋮
N

odd nos.

#iterations = # odd nos. from [1 N]

Q. How many odd nos. are there from [1 N] ?

N=7 [1 2 3 4 5 6 7] : 4

N=6 [1 2 3 4 5 6] : 3

$\frac{N}{2}$ $\frac{7}{2} = 3.\cancel{5} = 3$ $\frac{6}{2} = 3$

$\frac{7+1}{2} = \frac{8}{2} = 4$

$\frac{6+1}{2} = \frac{7}{2} = 3.\cancel{5} = 3$

$\frac{10+1}{2} = \frac{11}{2} = 5.\cancel{5}$
$= 5$

#iter: $\boxed{\frac{N+1}{2}}$

O(N)

Que.

```
int func(N) {
    s = 0
    for(i=0 ; i<=100 ; i++){
        s = s + i + i^2
    }
    return s
}
```

$i : [0 \quad 100]$

$100 - 0 + 1 = 101$

$\#\text{iterations} = 101$

$O(1)$

```
void func(N) {
    s = 0
    for( i=1 ; i*i <= N ; i++){
    }
}
```

$i * i <= N$

$i^2 <= N$

$i <= \sqrt{N}$

$i : [1 \quad \sqrt{N}]$

$\#\text{iteration} = \sqrt{N} - 1 + 1$

$= \sqrt{N}$

$O(\sqrt{N})$

```
void func(N){
    i = N
    while (i >= 1){
        i = i/2
    }
}
```

i

Before	Iteration	After	
N	1st	$N/2$	$N/2^1$
$N/2$	2nd	$N/4$	$N/2^2$
$N/4$	3rd	$N/8$	$N/2^3$
$N/8$	4th	$N/16$	$N/2^4$
		.	
		.	
		.	
	Kth	1	$N/2^K$

$$i: \underbrace{N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \rightarrow \cdots 1}_{K \text{ iteration}}$$

$$\frac{N}{2^K} = 1$$

$$N = 2^K$$

$$\log_2 N = \log_2 2^K$$

$$\log_2 N = K$$

$$\boxed{K = \log_2 N}$$

iterations: $\log_2 N$

$O(\log_2 N)$

Que.

```
void func(N) {
    for (i=0 ; i<=N ; i=i*2) {
        print (i)
    }
}
```

i

$0 \downarrow 0 \downarrow 0 \downarrow 0 \ldots$

#iteration : ∞

```
void func(N) {
    for (i=1 ; i<=N ; i=i*2) {
        print (i)
    }
}
```

$1 \xrightarrow{*2} N$

$N \xrightarrow{/2} N/2 \xrightarrow{/2} N/4 \ldots\ldots 1$

$N \xleftarrow{*2} N/2 \xleftarrow{*2} N/4 \ldots\ldots \xleftarrow{*2} 1$

#iterations : $\log_2 N$

$O(\log_2 N)$

$$\underbrace{1 \to 2 \to 4 \to 8 \to \cdots \to N}_{K \text{ iterations}}$$

$$1 \to 2^1 \to 2^2 \to 2^3 \to \cdots \to 2^K$$

$$N = 2^K$$

$$\boxed{K = \log_2 N}$$

Nested Loops

```
for (i = 1 ; i <= 10 ; i++) {
    for (j = 1 ; j <= N ; j++) {
        print (i*j)
    }
}
```

i	j	iterations
1	[1 N]	N
2	[1 N]	+ N
3	[1 N]	+ N
⋮		+
10	[1 N]	+ N

iterations: $10 \times N$

$\Downarrow$

$O(N)$

```
for (i = 1 ; i <= N ; i++) {
    for (j = 1 ; j <= N ; j++) {
        print (i*j)
    }
}
```

i	j	iterations
1	[1 N]	N
2	[1 N]	+ N
3	[1 N]	+ N
. . . .		+
N	[1 N]	+ N

iterations: N^2

$= O(N^2)$

```
for (i = 1 ; i <= 4 ; i++) {
    for (j = 1 ; j <= i ; j++) {
        print ("Monkey")
    }
}
```

i	j	iterations
1	[1 1]	1
2	[1 2]	+ 2
3	[1 3]	+ 3
4	[1 4]	+ 4

$\rightarrow O(1)$

iterations: 10

Sum of N Natural nos.

N = 4

$\frac{4 \times 5}{2} = 10$

```
(i = 0 ; i<N ; i++){
    (j=0 ; j<=i ; j++){
        print(i+j)
    }
}
```

i	j	Iterations
0	[0 0]	1 +
1	[0 1]	2 +
2	[0 2]	3
⋮		⋮ +
N-1	[0 N-1]	N

sum of N natural nos.

iterations: $\frac{N(N+1)}{2}$

$\frac{N^2+N}{2} = N^2+N$

$O(N^2)$

```
for(i=1; i<=N; i++) {
    for(j=1 ; j<=N; j=j*2){
        print(i*j)
    }
}
```

i	j	iteration
1	1 → N	$\log_2 N$ +
2	1 → N	$\log_2 N$ +
3	1 → N	$\log_2 N$ +
⋮		⋮ +
N	1 → N	$\log_2 N$

iterations: $N * \log_2 N$

$O(N \log_2 N)$

```
for ( i = 1 ; i <= 2^n ; i++ ) {

}
```

$i : [1 \quad 2^n]$

#iterations : 2^n

$O(2^n)$

Q.

```
for ( i = 1 ; i <= N ; i++ ) {
    for ( j = 1 ; j <= 2^i ; j++ ) {
        print( i * j )
    }
}
```

i	j	iterations
1	$[1 \quad 2^1]$	2^1
2	$[1 \quad 2^2]$	$+\ 2^2$
3	$[1 \quad 2^3]$	$+\ 2^3$
4	$[1 \quad 2^4]$	$+\ 2^4$
⋮		⋮
N	$[1 \quad 2^N]$	$+\ 2^N$

Sum GP $\quad \dfrac{a(r^N - 1)}{r - 1}$

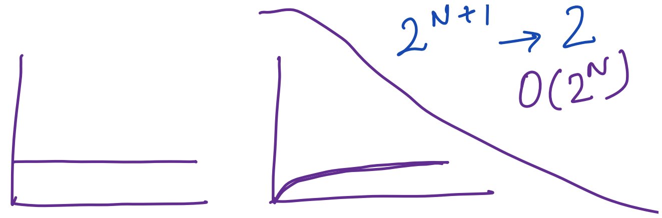
$a = 2$
$r = 2$
terms $= N$

$$= \frac{2(2^N - 1)}{2 - 1} = 2(2^N - 1)$$

$2^{N+1} - 2$

N

Order of Complexities



$2^{N+1} \to 2$

$O(2^N)$

$N = 32$

$$O(1) < \log_2 N < \sqrt{N} < N < N\log N < N\sqrt{N} < N^2 < 2^N < N! < N^N$$

$\log_2 32$	$\sqrt{32}$	32	32×5	32×5.65	$(32)^2$	2^{32}	
5	5.65	32	160	181	1024	4×10^9	$>> 4 \times 10^9$

Steps to find BIG OH Notation

1) Calc the no. of iterations
2) Neglect all lower order terms
3) Neglect constants from higher order term

$\sqrt{N}$ $\quad$ $\log_2 N$

2^{32}

$\sqrt{2^{32}}$ $\quad$ $\log_2 2^{32}$

$= 2^{16} >>> 32$

$N^8 \ll N^{40}$

#iterations: $\underbrace{4N^2}_{} + 3N + 1$

$\cancel{4}N^2 = \boxed{N^2}$

$\boxed{O(N^2)}$

#iterations: $\underbrace{4N^2}_{} + \cancel{3N} + \cancel{10^6}$

$4N^2$

$O(N^2)$

#iterations: $3N\sqrt{N} + 4\underbrace{\cancel{\log N}}_{} + \cancel{31\, N \log N}$

$\sqrt{N} > \log N$

$\cancel{3}N\sqrt{N}$

$O(N\sqrt{N})$