wednesday

Notes

Topic Name : Time Complexity – 1

+2

⇒ TLE : Time Limit Exceeded

Why TLE?

TODAY:

AGENDA:

* Time & Space complexities
* Asymptotic Analysis

  Big-O Notation
* Worst Case, Best Case, Avg Case.

Multiplication Tables

⇓

⇒ Iterations

17 – 20 quizzes

QUIZ: Pre-Requisites

Q1) Given N, how many steps to reach 1, at every step, we divide N by 2.

1

$N \to N/2 \to N/4 \to N/8 \to \cdots \to 1$

$\log_2(N)$

Q2) $[3, 10] \to 3, 4, 5, 6, 7, 8, 9, 10$

1 2 3 4 5 6 7 8

2

i) $[a, b] \Rightarrow b - a + 1$

a & b both are included

[ ] inclusion

$[3, 10] \Rightarrow 10 - 3 + 1 = 8$.

( ) exclusion.

range (a, b)

ii) $[a, b) \Rightarrow b - a$ $[3, 10) \Rightarrow 10 - 3 = 7$

iii) $(a, b) \Rightarrow b - a - 1$ $(3, 10) \Rightarrow 10 - 3 - 1 = 6$

NOTE:

Integer division

$9 // 2 = 4$

$10 // 2 = 5$

$11 // 2 = 5$

$9 / 2 = 4.5$

Take only integer part in integer division

Q3 Sum of N elements in A.P.

$a_N = a + (N-1)d$ — Nth term

a = first term

d = common difference.

1, 3, 5, 7, 9, 11, 13, 15, 17

(+2, +2)

$S_5 = 1+3+5+7+9$

$= 25$

$S_N = \frac{N}{2}[2a + (N-1)d]$ — Sum of first N terms

$S_5 = \frac{5}{2}[2*1 + (5-1)*2] = \frac{5}{2}[2+8] = \frac{10*5}{2} = 25$

$\log_a a^x = x$, $\log_2 2^{10} = 10$, $\log_2 2^n = n$

1. $1024 = 2^{10}$

Take $\log_2$ on both sides

$\log_2(2^{10}) = \log_2(1024)$

2. $\Rightarrow \log_2(1024) = 10$

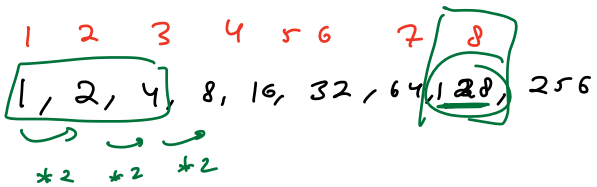
$\log_a(a^x) = x$

Base sets cancelled.

→ What is the power you need to raise for the number 2 to get 1024.

→ How many times you need to multiply 2 to get 1024!

GP : Geometric Progression



a = first term

r = common ratio

$a = 1$

$r = 2$

$a_n = a r^{n-1}$

$a_8 = 1 * 2^7$

$= 128$

$S_N = \frac{a(r^n - 1)}{r - 1}$

$S_3 = \frac{1(2^3 - 1)}{2 - 1}$

$= \frac{8-1}{1} = 7$

Equivalent Code as Quiz-4

```
s = 0
for (int i=1 ; i<=N ;i++){
    s = s+i ;
}
```

---

range(1, 100) => [1, 99]

range(1, 100, 2) 1, 3, 5, ---99

step increment. By default = 1

QUIZ: 4

Range last num is excluded

auto increments by 1

```
1  s = 0
2  for i in range(1, N+1):
3      s = s + i
```

start → 1, 2, 3, ... N ← end

i = 1, 2, 3, ... N

s = 1, 3, 6, ... - - -

[1, N] ⇒ N − 1 + 1 = N

QUIZ: 5

★ Bonus

Will be discussed in next class.

```
s = 0
for i in range(1, N+1, 2):
    s = s + i
```

QUIZ: 6

```
s = 0
for i in range(0, 101):
    s = s + i
```

Numbers [0, 101) — start, end.

101 − 0 = 101

[0, 100] ⇒ 100 − 0 + 1 = 101

QUIZ: 7

```
i = 1
while i*i <= N:
    i += 1
```

Loop Condⁿ

Executed only when $i^2 <= N$

upd ⁿ

N = 20

i	i*i = i^2
1	1
2	4
3	9
4	16
5	25 ✗

When does our loop stop?

$i^2 = N$

Take sqrt on both sides

$\Rightarrow \sqrt{i^2} = \sqrt{N}$

$\Rightarrow i = \sqrt{N}$

How many times did the loop run? ⇒ $\sqrt{20} = 4.6$

int = 4

Integer value

N = 25

i	i^2	After loop run
1	1	i=2
2	4	i=3
3	9	i=4
4	16	i=5
5	25	i=6
6	36	✗

QUIZ: 8

```
// Given N > 1
i = N
while (i > 1):
    i = i // 2
```

i Before	Total no. of iterations	i After current iteration $i = i//2$
$N > 1$	1	$i = N/2 = N/2^1$
$N/2 > 1$	2	$i = N/4 = N/2^2$
$N/4 > 1$	3	$i = N/8 = N/2^3$
$N/8 > 1$	4	$i = N/16 = N/2^4$

$2 \rightarrow 1$
$3 \rightarrow 1$
$4 \rightarrow 2 \rightarrow 1$
$5 \rightarrow 2 \rightarrow 1$
$6 \rightarrow 3 \rightarrow 1$
$7 \rightarrow 3 \rightarrow 1$
$8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

After K iterations → $i = \frac{N}{2^K}$

When does loop stop? After x iterations

$$i = \frac{N}{2^x} = 1$$

Find x

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

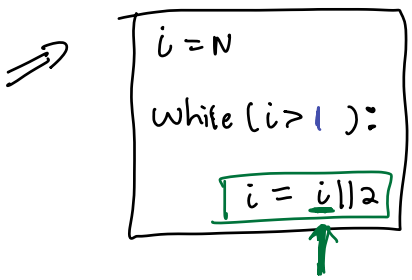
$$\Rightarrow N = 2^x$$

Take $\log_2$ on both sides

$$\Rightarrow \log_2 N = \log_2(2^x)$$

$$\log_2 N = x$$

After $\log_2(N)$ iterations i becomes 1



$N = 64 \Rightarrow$ 6 times

$\log_2(64) = 6$

i	Before	Total iterations	i After current iteration
64		1	$i = 64//2 = 32$
32		2	$i = 32//2 = 16$
16		3	$i = 16//2 = 8$
8		4	$i = 8//2 = 4$
4		5	$i = 4//2 = 2$
2		6	$i = 2//2 = 1$
1		X	

$1 > 1 \Rightarrow$ False

$i = 7$

i	#	Updated i
7	1	$7//2 = 3$
3	2	$3//2 = 1$
1		

$\log_2(7) = 2.$

QUIZ: 9

N ≥ 0

```
i = 0
while (i <= N):
    i = i*2
```

Infinitely

i Before	Iterations	i After (i = i*2)
i = 0	1	i = 0*2 = 0
0	2	i = 0*2 = 0
0	3	0
0	4	0
0	5	0
⋮	⋮	⋮
	∞	

5 min Break

~~Python~~

C++ / Java

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

Known termination

```
i = 1
while (i <= N) {
    i += 1;
}
```

Tricky Fn

Exact count not known.

Python

```
for i in range(1, N+1):
```

```
i = 1
while (i <= N):
    i += 1
```

~~i++~~

QUIZ: 10

$N = 20$

$1 \to 2 \to 4 \to 8 \to 16 \to 32$

$\lfloor \log_2 20 \rfloor = 4.3$

$4 + 1 = 5$

```
i = 1
while (i <= N)
    i = i * 2
```

i Before	Iterations	i After
		i = i * 2
$1 <= N$	1	$i = 1 * 2 = 2 = 2^1$
$2 <= N$	2	$i = 2 * 2 = 4 = 2^2$
$4 <= N$	3	$i = 4 * 2 = 8 = 2^3$
$8 <= N$	4	$i = 8 * 2 = 16 = 2^4$
16 :	5 :	:
32	6	

$1 \to 2 \to 4 \to 8 \to 16 \to 32$

reverse

$2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \quad 2^5$

$N \to N/2 \to N/4 \to \cdots \to 1$

$\log_2(N)$

After K iterations $\longrightarrow$ 2^K

When does this loop stop?

$N = 4 \Rightarrow 1 \to 2 \to 4 \to 8$ (1 2 3) — 3 times.

3 steps ✓

$2^1 \quad 2^2 \quad 2^3$

$N = 25$

$1 \to 2 \to 4 \to 8 \to 16 \to 32$ (1 2 3 4 5)

5 steps ✓

$\log_2(4) = 2$

$\log_2(25) = 4.\_\_$

$1 + \log_2(N)$

✓ After how many iterations — i becomes N. After x $\to$ N

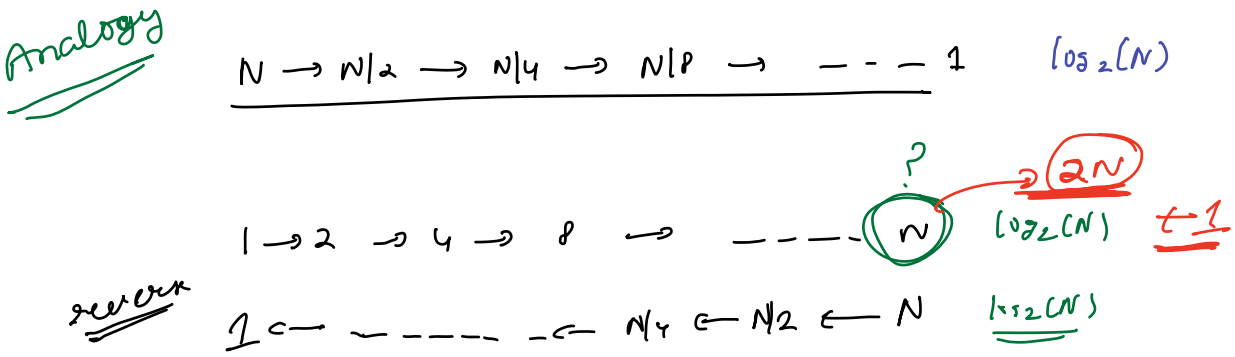
$2^x = N$

Take $\log_2$ both sides

$\Rightarrow x = \log_2 N$

✓ When does our loop stop? — 1 more after x iterations.

$\Rightarrow x + 1 = 1 + \log_2 N$ — Integer part



Nested Loops are here!

QUIZ: 11

$N > 0$

```
s = 0
for i in range(1, 11 ):
    for j in range(1, N+1):
        s += 1
```

No of times this statement is exec

⇓

gets executed $10 * N$ times

i	j	(Inner Loop) Total iterations
1	[1, N]	N times
2	[1, N]	N times
3	[1, N]	N time
⋮		
10	[1, N]	N time

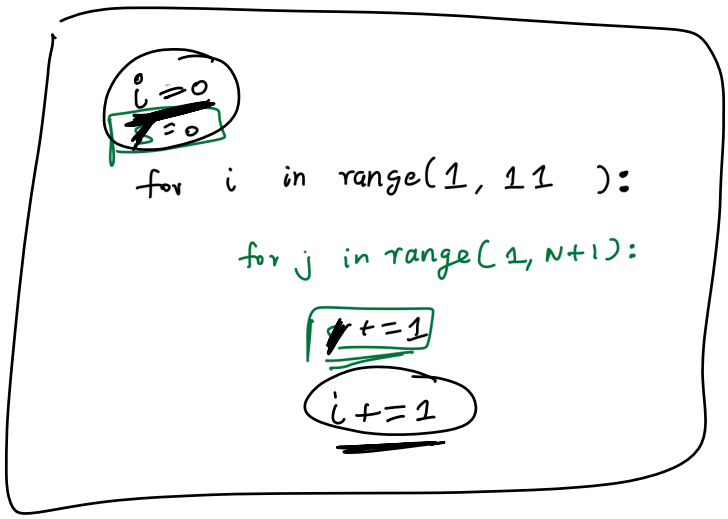
$N + N + \ldots + N$ (10 times)

$= 10N$

Outer loop Iterations : $[1, 10] \Rightarrow 10$

Inner loop Iterations : $10 * N$

↑ No of times inner loop runs for 1 single outer loop iteration



HW

Try what happens

QUIZ: 12

```
s = 0
for i in range(0,N):
    for j in range(0,N):
        s += 1
```

$[0, N) = N - 0 = N$

i	j	Inner Loop Total iterations
0	$[0, N-1]$	N
1	$[0, N-1]$	N
2	1	
	1	
	1	
	1	
$N-1$	$[0, N-1]$	N

Outer loop Iterations = N

Inner loop Iterations for 1 single outer loop $= N$

N iterations.

Total inner loop iterations $= N * N = N^2$

$N + N + N + \dots + N = N^2$.

QUIZ: 13

```
s = 0
for i in range(0, N):
    for j in range(0, (i+1)):
        s += 1
```

$i : [0, N) \Rightarrow$

i	j	Total iterations
0	[0, i] [or 0]	0−0+1 = 1
1	[0, 1]	1−0+1 = 2
2	[0, 2]	2−0+1 = 3
3	[0, 3]	3−0+1 = 4
⋮		⋮
N−1	[0, N−1]	(N−1)−0+1 = N

N iterations for outer loop.

range(0, 0) = [ ]

range(0, 1) = [0]

[0, 0]

[0, 1)

range(2, 2) = [ ]

Total inner loop iteration = $1 + 2 + 3 + 4 + \cdots + N$

---

① A.P. $a = 1$, $d = 1$

$$S_N = \frac{N}{2}[2a + (N-1)d] = \frac{N}{2}[2 + (N-1)1]$$

$$= \frac{N}{2}[N+1] = \frac{N^2+N}{2}$$ (C)

② $S = 1 + 2 + 3 + \cdots + N$

$S = N + (N-1) + (N-2) + \cdots + 1$

$2S = (N+1) + (N+1) + (N+1) + \cdots (N+1) = (N+1)(N)$

$\Rightarrow S = \frac{(N+1)N}{2}$

QUIZ: 14 ★ Bonus

```
s = 0
for i in range(1, N+1, 2):
    for j in range(1, i+1):
        s += 1
```

Will be discussed in the next class.

QUIZ: 15 * Bonus

```
for i in range(1, N+1):
    j = 1
    while j <= N:
        j = j * 2
```

Will be discussed in the next class.

QUIZ: 16

HW

```
s = 0
for i in range(1, 2**n+1):
    s += 1
```

```
s = 0
for i in range(1, N+1):
    for j in range(1, 2**i +1)
        s += 1
```

Doubts

→ $[0,1] = 0, 1$

$[0,1) = 0$ ⇐ range $(0,1)$

$(0,1) =$ empty.

→ Generalization

→ Cracking the Coding Interview

Elements of Programming Interviews.

$N = 30$

```
i = N
while i > 1 :
    i = i/2
```

Normal division vs Integer division.

$i = 30$	30	$\log_2(N)$
15	15	
7.5	7	
3.75	3	
1.875	1	For sure it will reach 1
0.9__	0	

$\log_2(N) + 1$ ⇐

* Code not giving expected output

1 - 2 hr

1- Debug by putting some print statements.

2- Try to do a dry run (manual run) on small input.

Try to contact TA.

---

* Not getting the logic

1. Try to read some hint/part of solution
2. Try to fill the missing parts.
3. Try to ask your peers.
   - Slack
   - Whatsapp.
4. TA
