Log $\Rightarrow$ inverse function to exponentiation.

$\log_b x \Rightarrow$ to what power should b be raised to reach x.

$\log_b n = k$

$$\boxed{b^k = n}$$

$\log_2 4 = 2$

$2^2 = 4$

$\log_3 81 = 4$

$3^4 = 81$

Ex1 $\log_2 8 \Rightarrow 3$

Ex2 $\log_2 2^{10} \Rightarrow 10$

$\log_2 n \Rightarrow k \Rightarrow$ no of times n needs to divided by 2 to reach 1.

Ex1 $\log_2 16 \Rightarrow 4$

Ex2 $\log_2 32 \Rightarrow 5$

$\log_2 n \Rightarrow k \Rightarrow$ How many times should I multiply 1 by 2 to reach n.

How many numbers in a range.

$[3, 10] = 8$

3, 4, 5, 6, 7, 8, 9, 10

$$\boxed{[a, b] \Rightarrow b - a + 1}$$

Both numbers are inclusive!

Airthmetic Progression

- Sequence of numbers where difference between consecutive terms is constant.

- Constant difference is known as common difference (d)

Ex1 : 1 , 4 , 7 , 10 , 13 , 16

Generalize : $a , a+d , a+2d , a+3d \ldots a+(n-1)d$

n^{th} term $\Rightarrow$ $a + (n-1)d$

Sum of 1^{st} n terms $= \frac{n}{2}\left[2a + (n-1)d\right]$

Ex1 : 1 , 5 , 9 , 13

$a = 1 , d = 4 , n = 4$

$\frac{4}{2}\left[2(1) + (4-1)4\right]$

$2 \times [14] = 28$

Geometric Progression

→ Sequence of numbers, where the next term can be found by multiplying the previous term by a fixed number.

↓ Common Ratio $[r]$

Ex: 2, 4, 8, 16, 32

Ex 2: 3, 12, 48, 192, 768 -- ---

Ex 3: 1, $1/2$, $1/4$, $1/8$, $1/16$ — . . .

Generalize: a, ar, ar^2, ar^3

n^{th} term $= ar^{n-1}$

Sum of 1st n terms $= \dfrac{a(r^n-1)}{r-1}$

Special Case: $\underline{r < 1}$ & series is infinite.

then $r^n \Rightarrow 0$.

Sum $\Rightarrow \dfrac{a}{1-r}$.

Q_1

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

$\Rightarrow i : [1, n]$

$n - 1 + 1 = n \Rightarrow O(n)$

Q_2

```
for (int i = 0 ; i <= 100 ; i++) {
        ____
}
```

$i : [0, 100]$

$= 100 - 0 + 1$

$\Rightarrow 101 = O(1)$

$\Downarrow$

time complexity is constant

Q_3 for (int i = 1 ; i ≤ n ; i += 2) {

3

$n \Rightarrow 4 \Rightarrow 2 = n/2$

$n \Rightarrow 5 \Rightarrow 3 = n/2$

$$\boxed{\left(\frac{n+1}{2}\right)}$$

$n = 4 \Rightarrow 5/2 \Rightarrow 2$

$n = 5 \Rightarrow 6/2 \Rightarrow 3$

→ Integer Divisor.

$$\frac{n+1}{2} \Rightarrow \frac{n}{\cancel{2}} + \cancel{1/2}$$

$$= O(n)$$

Q_4 for (int i = 1 ; i * i ≤ n ; i++) {

}

$i^2 \le n$

$i \le \sqrt{n}$

$i = [1, \sqrt{n}]$

$\Rightarrow \sqrt{n} - 1 + 1$

$\Rightarrow \sqrt{n}$

$\Rightarrow O(\sqrt{n})$

Q5

```
i = n
while (i > 1) {
    i = i/2
}
```

$i = n \downarrow n/2 \downarrow n/4 \downarrow n/8 \; \vdots \; 1$

no of iteration $= \log_2 n$

$= O(\log n)$

Q6

```
for (int i=0 ; i <= n ; i=i*2) {
}
```

$i = 0 \downarrow 0 \downarrow 0 \downarrow 0$

$\boxed{\text{infinite}}$

TC $\Rightarrow$ O(infinity)

$\underline{Q7}$ for (int i=1; i <= n; i = i*2) {

}

i = 1, 2, 4, 8, 16, 32 . . . . . n

$\log N$

$= O(\log N)$

10:39

NESTED LOOPS !

Q1 for (int i=1 ; i ≤ 10 ; i++) {

for (int j=1 ; j ≤ n ; j++) {

}

}

i	j	Total iteration
1	[1, n]	n
2	[1, n]	n
3	[1, n]	n
⋮		
10	[1, n]	n
10		10n

O(n) ⇒ 10 + 10n

```
int j = 0
for (int i = 0; i < 10^5; i++) {
    while (j < 100) {
        j++;
    }
}
```

i	j	Total
0	100	100
1	0	0
2	0	0
3	0	0
⋮		
$10^5 - 1$	0	0

$O(1)$

$10^5 + 100$

Q2

```
for (int i=1 ; i<=n ; i++) {
    for (int j=1 ; j<=n ; j++) {
        ________
    }
}
```

i	j	
1	[1, n]	n
⋮		⋮
n		n

$\underline{n} + \underline{n^2}$

$O(n^2)$

Q_3 for (int i = 1 ; i ≤ n ; i++) {

for (int j = 1 ; j ≤ n ; j = j * 2) {

}

}

i	j	Total
1	log n	log n
2	log n	log n
⋮		
n	log n	log n

$n + n\log n$

$O(n\log n)$

Q2 for (int i = 1 ; i ≤ n ; i++) {

for (int j = 1 ; j ≤ (2^i) ; j++) {

}

}

H.W

Hint

G.P

How to calculate Big O from the number of iterations!

1) Ignore lower order terms

2) Ignore constants

Q1 no of iterations $\Rightarrow 4n^2 + 3n + 2$

1st Step : Ignore lower order terms

$$4n^2 + \cancel{3n} + \cancel{2}$$

2nd Step : Ignore Constants.

$$\cancel{4}n^2$$

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

Q_2 no of iterations $\Rightarrow 3n\sqrt{n} + 4\log n + 3n\log n$

1st Step : Ignore lower order terms

$3n\sqrt{n} + \cancel{4\log n} + \cancel{3n\log n}$

n	$n\sqrt{n}$	$n\log n$
2^{32}	2^{48}	2^{37}
2^{64}	2^{96}	2^{70}

32

2nd Step : Ignore Constants.

$\cancel{3} n \cdot \sqrt{n}$

$O(n\sqrt{n})$

Q

$\sqrt{n} + \log n$

is O(1) less than O(n)

Doubts

```
for (int i=0 ; i<n ; i++) {
        if (i == 2)
              break;
}
```

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

}
```

$i = 1, 3, 5, 7 \ldots\ldots n$

$a + (x-1)d \Rightarrow n$

$1 + (x-1)2 \Rightarrow n$

$$(x-1)2 \Rightarrow n-1$$

$$x-1 \Rightarrow \frac{n-1}{2}$$

$$x \Rightarrow \frac{n-1}{2} + 1 = \frac{n+1}{2}$$

```
# for(int i=0 ; i<n ; i++) {
  ======          n=10^3      ] n
}

for (int i=0 ; i<m ; i++) {
  ======          m=10        ] m
}
```

$$n+m$$

$$O(n+m)$$

```
for (int i=0; i<=n; i++) {
    for (int j=0; j<i; j++
    ____________
    ________
}
```

i	j	[illegible]
0	0	0
1	1	1
2	2	2
3	3	3
n	n	n

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

$$O(n^2)$$

$$\left(2^0+2^1+2^2 \cdots - 2^n\right)$$

$$\simeq 2\left(2^n - \cdots\right)$$

$$= \underline{\underline{O(2^n)}}$$