→ Time complexity

Generally in coding contests : no. of iterations in 1 sec => $10^7 - 10^8$

∴ 10^7 → 1 sec

10^8 → 10 sec

10^9 → 100 sec

10^{10} → 1000 sec

10^{11} → 10000 sec

→
```
INT X ;           // O(1)
INT SUM = 0 ;     // O(1)
SUM = X*X ;       // O(1)
```

No. of iterations = 3   ∴ O(1) + O(1) + O(1)
=> O(3) ≈ O(1)

→
```
INT X ;                 // O(1)
INT SUM = 0 ;           // O(1)
SUM = X*X ;             // O(1)
INT N;                  // O(1)
CIN >> N;               // O(1)
FOR (INT i=0; i<n; ++i) {
      INT Y;            // O(n)
      Y = 5;            // O(n)
      Y++ ;             // O(n)
   }
}
```

Total time complexity : O(5) + O(3n) ≈ O(n)

→
```
FOR (INT i=0 ; i<n; ++i) {
      CIN >> TEMP;                 // O (n)
}
FOR (INT i=0; i<n; ++i) {
      CIN >> TEMP2;                // O(n)
}
FOR (INT i=0; i<n; ++i) {
      CIN >> TEMP3;                // O(n)
}
```

Total time complexity : $O(n) + O(n) + O(n) \approx O(n)$

```
→ FOR (INT i=0; i< n; ++i) {
      FOR (INT i=0; i<n; ++i) {
          CIN >> a[i]                    // O(n^2)
      }
  }
```

Total time complexity : $O(n^2)$

```
→ INT n;
  CIN >> n;          // let n = 10^5
  WHILE (n > 0) {
      n = n/2;
  }
```

Total time complexity : eg: n = 10

10 – 5 – 2 – 1 – 0
↑ ↑ ↑
3 Iterations

If n = 20

20 – 10 – 5 – 2 – 1 – 0
↑ ↑ ↑ ↑
4 Iterations

If n = 100

100 – 50 – 25 – 12 – 6 – 3 – 1 – 0
↓ ↓ ↓ ↓ ↓ ↓
6 Iterations

Let a is the no. of times we can divide n to make it zero (0).

$\therefore 2^a = n$

$\log_2(2^a) = \log_2(n)$

$a \log_2 2 = \log_2(n)$

$\therefore \boxed{a = \log_2(n)}$ → This is the best time complexity we can have.

e.g if $n = 10^5$

$\therefore \log_2(10)^5$

$= 5 \times \log_2(10)$

$= 5 \times 3$

$= \underline{\underline{15 \text{ iterations}}}$