Time Complexity

↳ Basically the amt of time it takes to execute something

Eg:

```
int count = 0 ;
for (int i=0; i<N ; i++){
    for (int j=0; j<i; j++){
        count++;
    }
}
```

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

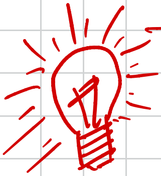
$$1+2+3+\ldots+N = \frac{N(N+1)}{2}$$

So basically time taken to run the above code = $\boxed{N^2+3N+1}$

Order Notations

↳ Types:

- → $O(f(n))$ → upper bound
- → $\Omega(f(n))$ → lower bount
- → $\theta(f(n))$ → Sandwhich of O & Ω



Sequence	Nesting
↓	↓
addition	multiplication

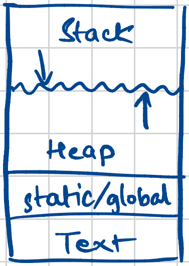
↳ JUST KEEP IN MIND

$n \le 400$	$O(n^3)$
$n \le 7500$	$O(n^2)$
$n \le 10^5$	$O(N\sqrt{N})$
$n \le 5\times10^5$	$O(n\log n)$
$n \le 5\times10^6$	$O(N)$
$n \le 10^{12}$	$O(\sqrt{N}\log N)$, $O(\sqrt{N})$
$n \le 10^{18}$	$O(\log^2 N)$, $O(1)$, $O(\log N)$

AMORTIZATION ?

Memory Complexity

In cpp,



$\underbrace{\text{arr}[n][N]}$

↳ $O(N^2)$

NP-HARD → You should know what these problems are. TO LEAVE THEM !

↳ Problems not solvable in polynomial time.

Finding time complexities for RECURSIVE PROBLEMS

Master Theorem

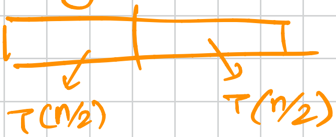
↳ Requirement :

$$\underline{T(n)} = aT\left(\frac{n}{b}\right) + c$$

How much time will it take to solve instance of size `n'.

applicable ✓

Let say MERGE SORT



So,

$$T(n) = 2T\left(\frac{n}{2}\right) + O(N)$$

Calculating for merge sort,

Step 1 : Note a, b, c

Step 2 : Calculate $\left(n^{\log_b a}, c\right)$

$\Rightarrow O(n^{\log_2 2}), O(N) \Rightarrow O(N), O(N)$

Step 3 : if $O(n^{\log_b a})$ & c → SAME

↳ THEN → ans is clogn

if $O(n^{\log_b a}) > c$

↳ THEN → ans is $O(n^{\log_b a})$

else

↳ ans is $O(c)$

∴ for merge sort → TC → $O(n\log n)$

Examples

↳ ① $T(n) = 2T(n/2) + O(n^2)$

Now, $n^{\log_2 2} < n^2$

∴ TC → $O(n^2)$

② $T(n) = 2T\left(\frac{n}{2}\right) + O(1)$

$n^{\log_2 2} > 1$

⇒ TC → $O(N)$

③ $T(n) = 8T\left(\frac{n}{2}\right) + \frac{n^3}{\log n}$

↳ Master Th. doesn't apply when you don't have a POLYNOMIAL EXPRESS^N

But still If you have to comment on TC.

↳ $c \log n$

$= O(n^3 \log n)$ → leaving out the log part.