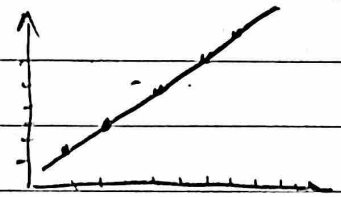
TIME COMPLEXITY BIG O NOTATION

TIME COMPLEXITY = A WAY OF SHOWING HOW THE RUNTIME OF THE FUNCTION INCRESES AS THE SIZE OF THE INPUT INCRESES.

- **LINEAR TIME** $O(n)$

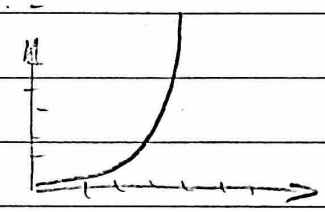
$T = an + b = O(n)$

1) FIND THE FASTEST GROWING TERM
2) TAKE OUT THE COEFFICIENT



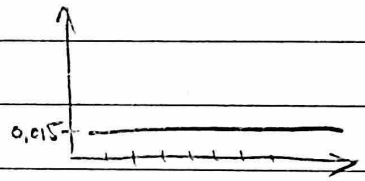
- **QUADRATIC TIME** $O(n^2)$

$T = cn^2 + dn + e = O(n^2)$



- **CONSTANT TIME** $O(1)$

$T = c = 0{,}015 = 0{,}015 \times 1 = O(1)$



CONSTANT

```
FUNCTION FUNC(){
    TOTAL = 0;  -> O(1)
    RETURN TOTAL; -> O(1)
}
```

$T = O(1) + O(1) = c1 + c2 = c_3 = c_3 \times 1 = O(1)$

LINEAR

```
FUNCTION FUNC(ARRAY){
    TOTAL = 0;   -> O(1)
    FOREACH(EL IN ARRAY){
        TOTAL =+ EL;  -> O(1)
    }
    RETURN TOTAL  -> O(1)
}
```

$T = O(1) + O(1) \times n + O(1) = c_4 + c_5 \times n = O(n)$

QUADRATIC

```
ARRAY_2D =
=[[1,2,3],
  [4,5,6],
  [7,8,9]]
```

```
FUNCTION FUNC(ARRAY_2d){
    TOTAL = 0;   -> O(1)
    FOREACH(ROW IN ARRAY_2D){
        FOREACH(EL IN ROW){
            TOTAL =+ EL; -> O(1)
        }
    }
    RETURN TOTAL; -> O(1)
}
```

$T = O(1) + O(1) \times n^2 + O(1) = c_6 + c_7 \times n^2 = O(n^2)$