



## Kahoot: Complejidad algorítmica

1. En el infinito  $O(\log n)$  tarda lo mismo que  $O(n)$   
verdadero  
**falso**
2. En el infinito  $O(74n)$  tarda lo mismo que  $O(n)$   
**verdadero**  
falso
3. Podemos representar la tasa de crecimiento de un algoritmo tanto como  $T(N)$  como  $F(N)$   
verdadero  
**falso**
4. La notación  $T_{\max}(N)$  hace referencia al peor caso de un tiempo de compilación  
verdadero  
**falso**
5. ¿Cómo se expresa el teorema maestro?  
 $T(n) = T(n/B) + A * f(n)$   
 **$T(n) = T(n/B) * A + f(n)$**   
 $T(n) = T(n/A) + B * f(n)$   
 $T(n) = T(n/A) * B + f(n)$
6. Ordenar de mayor a menor  

$O(n^4)$	3
$O(\pi)$	1
$O(n!)$	4
$O(n \log_3(n))$	2
- 7.Cuál es la complejidad del siguiente algoritmo?

```

void alimentar_pokemon(pokemon_t* pokemon, comida_t* comida){
    if(!pokemon || !comida || !comida->pokehabas)
        return

    for(int i=0, i<comida->cant_pokehabas, i++){
        comer_pokehaba(pokemon, comida->cant_pokehabas[i]); // <- O(1)
    }

    dormir_siestita(pokemon); // <- O(5)
}

```

$O(\text{comida} \rightarrow \text{cant\_pokehabas} - 1)$

$O(\text{comida} \rightarrow \text{cant\_pokehabas})$

$O(\text{comida} \rightarrow \text{cant\_pokehabas}) + O(5)$

$O(1) + O(5)$

8. "Big Omega" describe el comportamiento de una función en el límite verdadero

falso