

Complejidad algorítmica

1. La notación $T_{\max}(N)$ hace referencia al peor caso de un tiempo de compilación
verdadero
falso
2. Escribir $T(N)$ es equivalente a escribir $F(N)$
sí
mm sospechoso
depende del valor de N
uno es para el tiempo de ejecución y el otro para el de compilación
3. $O(n/2)$ tarda lo mismo que $O(3n)$
siempre
solo en el infinito
nunca
4. Ordenar de menor a mayor complejidad
 $O(n \log(n)) \rightarrow 3$
 $O(n) \rightarrow 2$
 $O(n^3) \rightarrow 4$
 $O(3) \rightarrow 1$
5. Como 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. Si $f(n)$ es mayor que $n^{\log_b(a)}$ entonces $T(n)$ es...
 $O(f(n) * n)$
 $O(f(n))$
 $O(n)$
 $O(\log_b(a))$
7. Cual es la complejidad para este algoritmo?

```
int una_funcion(int contador){  
    if(contador == MAX_CONTADOR){  
        return ERROR;  
    }  
    while(contador < TOPE_CONTADOR){  
        imprimir_valor(contador);  
        contador += 2;  
    }  
    return EXITO;  
}
```

$O(\text{TOPE_CONTADOR})$
 $O(\text{TOPE_CONTADOR}/2)$
 $O(\text{TOPE_CONTADOR} * \text{contador})$
 $O(\text{TOPE_CONTADOR}) + O(1)$

8. El tiempo de ejecución del siguiente algoritmo es la multiplicación de lo que tarda cada cachito

```
if(una_condicion){  
    un_cachito_de_codigo();  
}  
else{  
    otro_cachito();  
}
```

verdadero

falso