

1. Fragmento de código condicional:

$$f(x) = \begin{cases} O(1 + 1 + 1 + 1), & num > 10 \\ O(1 + 1 + 1 + 1 + 1 + 1 + 1), & num \leq 10 \end{cases}$$

$$f(x) = \begin{cases} O(4), & num > 10 \\ O(6), & num \leq 10 \end{cases}$$

$$f(x) = O(1)$$

2. Fragmento de código condicional:

$$f(x) = \begin{cases} O(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1), & num > 10 \\ O(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1), & num \leq 10 \end{cases}$$

$$f(x) = \begin{cases} O(10), & num > 10 \\ O(7), & num \leq 10 \end{cases}$$

$$f(x) = O(1)$$

3. Fragmento de código de ciclo:

$$f(x) = O(n) + O(n * m) + O(n * m * 3)$$

$$f(x) = (n * m * 3)$$

4. Fragmento de código de ciclo:

$$f(x) = \begin{cases} O(n * m), & n \neq 0 \\ O(n * 10), & n = 0 \end{cases}$$

$$f(x) = O(n * m)$$

5. Fragmento de código condicional:

$$f(x) = \begin{cases} O(1 + 1 + 1 + 1 + 1 + 1 + 16), & num > 10 \\ O(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1), & num \leq 10 \end{cases}$$

$$f(x) = \begin{cases} O(22), & num > 10 \\ O(7), & num \leq 10 \end{cases}$$

$$f(x) = O(1)$$

6. Fragmento de código:

$$f(x) = O(1 + n + n)$$

$$f(x) = O(n)$$

7. Búsqueda binaria iterativa:

Este algoritmo divide por 2 en cada iteración la cantidad de pasos posibles, por lo que se define como el logaritmo en base 2 de la cantidad de pasos totales posibles al comienzo

$$f(x) = \log_2 n$$

$$f(x) = \log n$$

8. Fragmento de código (mezcla de listas):

$$f(x) = O(1 + 1 + n + m + 1 + n + 1 + m + 1)$$

$$f(x) = O(2n + 2m + 5)$$

$$f(x) = O(n + m)$$

9. Fragmento de código:

$$f(x) = O(n * (1 * j * (1 + 1)) + 1 + n * (1 + 1))$$

$$f(x) = O(n * j + n)$$

10. Fragmento de código:

$$f(x) = O((n-1) * (1 + 1 + 1 + 1 + 1 + (n-1) * (1 + 1 + 1) + 1 + 1))$$

$$f(x) = ((n-1) * (7 + (n-1) * 3))$$

$$f(x) = O(n^2)$$

11. Código de multiplicación de dos matrices M1[n x m] y M2[m x o]:

$$f(x) = O(n * o * (1 + m * 1 + 1))$$

$$f(x) = O(n * o * m)$$

12. Código de suma de dos matrices de [n x n]:

$$f(x) = O(n * n * 1)$$

$$f(x) = O(n^2)$$

13. Código para calcular la traza de una matriz cuadrada:

$$f(x) = O(2 + n * 1 + 1)$$

$$f(x) = O(n)$$

14. Código para calcular el determinante de una matriz cuadrada de [3 x 3], regla de Sarrus:

$$f(x) = O(1 + m * (1 + 1 + m * (1 + 1 + 1 + 1) + 1) + (m - 1) * (1 + 1 + m * (1 + 1 + 1 + 1) + 1))$$

$$f(x) = O(1 + m * (3 + m * 4) + (m - 1) * (3 + m * 4))$$

$$f(x) = O(m^4)$$

15. Función para determinar si un número es primo:

$$f(x) = \begin{cases} O(1 + 1), & n \leq 1 \\ O(1 + 1 + (n-1) * (1 + 1) + 1 + 1 + 1 + 1), & n > 1 \end{cases}$$

$$f(x) = \begin{cases} O(2), & n \leq 1 \\ O((n-1) * (2) + 6), & n > 1 \end{cases}$$

$$f(x) = O(n)$$

16. Función factorial iterativa:

$$f(x) = O(1 + n * 1 + 1)$$

$$f(x) = O(n)$$

17. Función Fibonacci iterativa:

$$f(x) = \begin{cases} O(1 + 1), & n = 0 \\ O(1 + 1), & n = 1 \\ O(1 + (n-1) * (1 + 1 + 1) + 1), & n! = 0 \text{ y } ! = 1 \end{cases}$$

$$f(x) = O(1 + 1 + n)$$

$$f(x) = O(n)$$