



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

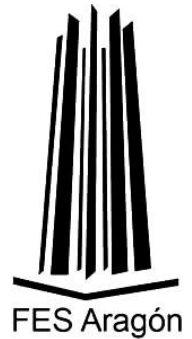
INGENIERÍA EN COMPUTACIÓN

DISEÑO Y ANÁLISIS DE ALGORITMOS

GRUPO 1558 2026 – 1

PROFESOR JESÚS HERNÁNDEZ CABRERA

ALUMNO PEDRO CÉSAR JUÁREZ NÚÑEZ



TAREA 3. Análisis de Algoritmos

INSTRUCCIONES:

1. Encontrar el $T(n)$ de cada uno de los 5 algoritmos (del pdf).
2. Aplicar las 5 reglas para la reducción y encontrar el Big O de cada una.

CÓDIGO 1.

```
def uno(n):  
    s = 0      # T(n) = 1  
    for i in range(n+1):      # T(n) = n  
        basura = i      # T(n) = 1  
        s += i      # T(n) = 1  
        i = i      # T(n) = 1  
    return s      # T(n) = 1
```

$$T(n) = 3(n + 1) + 2 = 3n + 3 + 2 = 3n + 5$$

$T(n) = n$

CÓDIGO 2.

```
def dos(n):  
    cnt = 0      # T(n) = 1  
    for _ in range(n):      # T(n) = n  
        j = n      # T(n) = 1  
        j = j      # T(n) = 1  
        while j > 1:      # T(n) = log_2(m)  
            j //= 2      # T(n) = 1  
            j = j      # T(n) = 1  
            cnt += 1      # T(n) = 1  
            cnt = cnt      # T(n) = 1  
    print("fin del for")      # T(n) = 1  
    return cnt      # T(n) = 1
```

$$T(n) = 4(\log_2(n)) * 2n + 3 = 8 * n * \log_2(n) + 3$$

$$T(n) = n * \log_2(n)$$

CÓDIGO 3.

```
# a es una lista

def tres(a):
    cnt = 0      # T(n) = 1
    n = len(a)   # T(n) = 1
    for i in range(n):      # T(n) = n
        for j in range(i+1):      # T(n) = m
            cnt += 1      # T(n) = 1
    return cnt      # T(n) = 1
```

$$T(n) = 3 + n * m$$

$$T(n) = nm$$

CÓDIGO 4.

```
def cuatro(a):
    n = len(a)      # T(n) = 1
    total = 0       # T(n) = 1
    n = (n * 2) / 2  # T(n) = 1
    for i in range(n):      # T(n) = n
        basura = 1      # T(n) = 1
        nada = 0        # T(n) = 1
        i = i            # T(n) = 1
        for j in range(n):      # T(n) = n
            basura2 = 1      # T(n) = 1
            otra_cosa = 2     # T(n) = 1
            j = j            # T(n) = 1
            for k in range(n):      # T(n) = n
                total = total      # T(n) = 1
                k = k              # T(n) = 1
                j = j              # T(n) = 1
                total += a[i] + a[j] + a[k]      # T(n) = 1
    return total      # T(n) = 1
```

$$T(n) = 4 + 3n * 3n * 4n = 36n^3 + 4$$

$$T(n) = n^3$$

CÓDIGO 5.

```
def operations(numbers):  
    results = []          # T(n) = 1  
  
    for number in numbers:    # T(n) = n  
        count = 0          # T(n) = 1  
        while number >= 1:    # T(n) = log_2(n)  
            number /= 2      # T(n) = 1  
            count += 1       # T(n) = 1  
            results.append(count)  # T(n) = 1  
  
    return results          # T(n) = 1
```

$$T(n) = 2 + 2n * 2\log_2(n) = 2 + 4n * \log_2(n)$$

$$T(n) = n * \log_2(n)$$