Análisis formal de complejidad para `contar_por_digito_volumen`

Sea n el número de elementos en el arreglo `arr`.

1. Planteamiento y desarrollo de $T(n)$
Analicemos cada ciclo:

- Declaración de arreglos: t_0 (constante)
- Primer ciclo: `for (int i = 0; i < n; i++)` (llamémosle t_1 por iteración)

$$T_1(n) = \sum_{i=0}^{n-1} t_1 = nt_1$$

- Segundo ciclo: `for (int i = 1; i < 10; i++)` (t_2 por iteración)

$$T_2 = \sum_{i=1}^{9} t_2 = 9t_2$$

- Tercer ciclo: `for (int i = n-1; i >= 0; i--)` (t_3 por iteración)

$$T_3(n) = \sum_{i=0}^{n-1} t_3 = nt_3$$

- Cuarto ciclo: `for (int i = 0; i < n; i++)` (t_4 por iteración)

$$T_4(n) = \sum_{i=0}^{n-1} t_4 = nt_4$$

Sumando todos los términos:

$$T(n) = t_0 + nt_1 + 9t_2 + nt_3 + nt_4$$

$$T(n) = t_0 + 9t_2 + n(t_1 + t_3 + t_4)$$

Redefinamos $c_1 = t_1 + t_3 + t_4$, $c_2 = t_0 + 9t_2$:

$$T(n) = c_1 n + c_2$$

---

2. Calculando los casos
Mejor caso: Incluso si $n = 0$, los ciclos internos (excepto el segundo) no se ejecutan, sólo queda el costo de inicialización y el segundo ciclo:

$$T_m(0) = t_0 + 9t_2 = c_2$$

Por lo tanto, $T_m(n) \in \Theta(1)$ para $n = 0$. Para $n > 0$, $T_m(n) = c_1 n + c_2 \implies \Theta(n)$

Peor caso: El ciclo siempre recorre todos los n elementos, igual que el análisis anterior:

$$T_p(n) = c_1 n + c_2$$

Por lo tanto, $T_p(n) \in \Theta(n)$.

Caso promedio: La función siempre ejecuta los mismos ciclos, independientemente de los datos, así que:

$$T_{pr}(n) = c_1 n + c_2$$

Por lo tanto, $T_{pr}(n) \in \Theta(n)$.

---

3. Comprobación de cotas por límites

$$\lim_{n \to \infty} \frac{T(n)}{n} = c_1$$

Por lo tanto:

$$T(n) \in \mathcal{O}(n)$$

$$T(n) \in \Omega(n)$$

$$T(n) \in \Theta(n)$$

4. Resumen Final:

- Mejor caso: $\Theta(n)$
- Peor caso: $\Theta(n)$
- Caso promedio: $\Theta(n)$

Cada cota ha sido verificada con límites y se han desarrollado las sumatorias con los nombres reales del código.