Análisis formal de complejidad para `resolver_top_down`

Sean:

- n: cantidad de pedidos (`i` desde 0 hasta $n-1$)
- W: peso máximo (`peso_restante`)
- V: volumen máximo (`volumen_restante`)

1. Planteamiento de $T(n, W, V)$:

Como es una función de **programación dinámica con memoización** (tipo mochila multidimensional), el número de subproblemas únicos es $n \cdot W \cdot V$ y cada subproblema toma tiempo constante.

Peor caso: (Ningún subproblema está en la tabla)

$$T_p(n, W, V) = C_1 + \sum_{i=0}^{n-1} \sum_{w=0}^{W} \sum_{v=0}^{V} C_2$$

$$T_p(n, W, V) = C_1 + n \cdot (W+1) \cdot (V+1) \cdot C_2$$

Mejor caso: (Todos los subproblemas ya están memoizados, solo realiza la comparación y regresa el valor)

$$T_m(n, W, V) = C_1$$

Caso promedio: (Solo una fracción α de subproblemas no está memoizada)

$$T_{pr}(n, W, V) = C_1 + \alpha \cdot n(W+1)(V+1)C_2$$

donde $0 < \alpha < 1$

---

2. Demostración de cotas por límites

Peor caso:

$$\lim_{n,W,V\to\infty} \frac{T_p(n, W, V)}{nWV} = \lim_{n,W,V\to\infty} \frac{C_1 + n(W+1)(V+1)C_2}{nWV}$$

El término dominante es nWV, así que:

$$= C_2 \lim_{n,W,V\to\infty} \frac{(W+1)(V+1)}{WV} = C_2$$

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

Mejor caso:

$$\lim_{n,W,V\to\infty} \frac{T_m(n, W, V)}{1} = C_1$$

Por lo tanto, $T_m(n, W, V) \in \Theta(1)$.

Caso promedio:

$$\lim_{n,W,V\to\infty} \frac{T_{pr}(n, W, V)}{nWV} = \alpha C_2$$

Con $0 < \alpha < 1$, entonces $T_{pr}(n, W, V) \in \Theta(nWV)$.

---

3. Conclusión final:

- Mejor caso: $\Theta(1)$
- Peor caso: $\Theta(nWV)$
- Caso promedio: $\Theta(nWV)$

En el peor y caso promedio, la función tiene complejidad cúbica en los parámetros relevantes.