Soluciones - CP0

José Rafael Pérez Rivero C-122

29 octubre 2025

Problema 1: Las 12 Monedas

Solución

Dividimos las 12 monedas en tres grupos de 4: Grupo A, Grupo B, Grupo C.

Primera Pesada: A vs B

- **Caso 1:** $A = B \rightarrow$ La falsa está en C
- **Caso 2:** $A < B \rightarrow$ La falsa está en A (si es más ligera) o en B (si es más pesada)
- **Caso 3:** $A > B \rightarrow$ La falsa está en A (si es más pesada) o en B (si es más ligera)

Segunda Pesada (Caso $A = B$):

Pesamos 3 monedas de C contra 3 monedas buenas de A:

- Si se equilibran: La moneda restante en C es falsa (tercera pesada para determinar si es más pesada/ligera)
- Si $C < A$: Una de las 3 de C es más ligera
- Si $C > A$: Una de las 3 de C es más pesada

Tercera Pesada:

Con solo 3 monedas sospechosas, pesamos 1 vs 1:

- Si igualan: La tercera es falsa
- Si no: La más ligera/pesada según el caso anterior

Complejidad

- Número máximo de pesadas: 3
- Estrategia: **Búsqueda ternaria**

Problema 2: Torres de Hanoi

Solución Matemática

Definimos $T(n)$ como el número mínimo de movimientos para n discos:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ 2T(n-1) + 1 & \text{si } n > 1 \end{cases}$$

Resolviendo la recurrencia:

$$T(n) = 2^n - 1$$

Algoritmo Recursivo

```python
def hanoi(n, origen, destino, auxiliar):
    if n == 1:
        print(f"Mover disco 1 de {origen} a {destino}")
        return
    # Mover n-1 discos de origen a auxiliar usando destino como auxiliar
    hanoi(n-1, origen, auxiliar, destino)
    # Mover el disco n de origen a destino
    print(f"Mover disco {n} de {origen} a {destino}")
    # Mover n-1 discos de auxiliar a destino usando origen como auxiliar
    hanoi(n-1, auxiliar, destino, origen)

# Ejemplo para 3 discos
hanoi(3, 'A', 'C', 'B')
```

Explicación del Algoritmo

1. **Caso base:** Si solo hay 1 disco, moverlo directamente del origen al destino
2. **Paso recursivo:**
   - Mover $n-1$ discos del origen al auxiliar (usando el destino como auxiliar temporal)
   - Mover el disco n (el más grande) del origen al destino
   - Mover los $n-1$ discos del auxiliar al destino (usando el origen como auxiliar temporal)

Ejemplo para 3 discos

```
Mover disco 1 de A a C
Mover disco 2 de A a B
Mover disco 1 de C a B
Mover disco 3 de A a C
Mover disco 1 de B a A
```

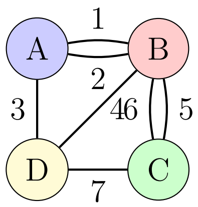
```
Mover disco 2 de B a C
Mover disco 1 de A a C
```

Análisis

- **Tiempo:** $O(2^n)$
- **Espacio:** $O(n)$ (pila de recursión)

Problema 3: Los Puentes de Königsberg

Modelado como Grafo



Teorema de Euler

Un grafo conexo tiene un **circuito euleriano** si y solo si:

$$\forall v \in V, \quad \deg(v) \text{ es par}$$

Aplicación al Problema

Grados de los vértices:

$$\begin{aligned} \deg(A) &= 3 \quad \text{(impar)} \\ \deg(B) &= 5 \quad \text{(impar)} \\ \deg(C) &= 3 \quad \text{(impar)} \\ \deg(D) &= 3 \quad \text{(impar)} \end{aligned}$$

Conclusión: No existe solución - Demasiados vértices de grado impar.

Problema 4: Los Jarrones

Solución Paso a Paso

Paso	Estado (5L, 3L)	Acción
0	(0, 0)	Inicio
1	(5, 0)	Llenar jarro de 5L
2	(2, 3)	Verter 5L en 3L
3	(2, 0)	Vaciar jarro de 3L
4	(0, 2)	Verter 5L en 3L
5	(5, 2)	Llenar jarro de 5L
6	**(4, 3)**	Verter 5L en 3L (hasta llenar)

Generalización

Para jarrones de capacidad a y b, queremos medir c litros. Solución existe si:

$$\gcd(a, b) \mid c$$

En nuestro caso: $\gcd(5, 3) = 1$ y $1 \mid 4 \rightarrow$ **Solución existe**.

Problema 5: Prisioneros y Sombreros

Estrategia Óptima

Los prisioneros acuerdan usar **código de paridad**:

- **Prisionero 1** (el último): Cuenta sombreros **negros** que ve
  - Si es **par**: Dice "BLANCO"
  - Si es **impar**: Dice "NEGRO"
- **Prisioneros** $2, \ldots, n$:
  - Escuchan todas las respuestas anteriores
  - Cuentan sombreros negros que ven + información de paridad
  - Deducen su propio sombrero con certeza

Ejemplo con 4 Prisioneros

Supongamos sombreros: [NEGRO, BLANCO, NEGRO, BLANCO]

1. **P1** ve: [?, BLANCO, NEGRO, BLANCO] $\rightarrow$ 1 negro (impar) $\rightarrow$ Dice "NEGRO"
2. **P2** oye "NEGRO", ve [?, NEGRO, BLANCO] $\rightarrow$ Deducen que tiene BLANCO
3. **P3** oye "NEGRO, BLANCO", ve [?, BLANCO] $\rightarrow$ Deducen que tiene NEGRO
4. **P4** oye todo $\rightarrow$ Deducen que tiene BLANCO

Análisis de Eficiencia

- **Prisionero 1**: 50 % de éxito
- **Prisioneros** $2, \ldots, n$: 100 % de éxito
- **Eficiencia total**: $n - 1$ salvados garantizados
- **Técnica**: Codificación por paridad XOR