Análisis y Diseño de Algoritmos
Ejercicios de práctica

- Demuestra que $3^n - 1$ es par para $n \geq 0$.
  Verificamos a mano para $n = 0$, teniendo $3^0 - 1 = 1 - 1 = 0$ que es par. Supondremos que es cierto para cierta n y verificamos por inducción para $n+1$. Evaluando para $n+1$ tenemos $3^{n+1} - 1 = 3(3^n) - 1 = 2(3^n) + (3^n - 1)$. Ambos sumandos son pares, por lo que la suma también es par.

- Demuestra que el siguiente algoritmo para calcular la resta de dos enteros no negativos es correcto:

  función RESTA($a, b \in \mathbb{N}$)
      si $b = 0$ **entonces**
          regresa a
      si no
          regresa Resta($a - 1, b - 1$)

  El caso base es $b = 0$ y observamos que $a - b = a - 0 = a$ que es lo que devuelve el algoritmo. Sean a, b los valores de la invocación original con $b > 0$. Supondremos que la llamada $Resta(a-1, b-1)$ es correcta, es decir que devuelve $(a - 1) - (b - 1)$. Nuestra función devuelve directamente lo que regresa la llamada recursiva. Observamos que $a - b = (a-1) - (b-1) = a - 1 - b + 1 = a - b$ que es lo que se deseaba demostrar.

- Demuestra que el siguiente algoritmo para calcular la resta de dos enteros no negativos es correcto:

  función RESTA($a, b \in \mathbb{N}$)
      mientras $b \neq 0$
          $a \leftarrow a - 1$
          $b \leftarrow b - 1$
      regresa a

  El único lugar donde a, b son modificados es dentro del ciclo. Sean a, b los valores originales de los argumentos y a_t, b_t los valores que tienen durante la t-ésima evaluación de la condición del ciclo (independientemente de si la condición fue verdadera o no). La invariante que proponemos es $a_t = a_0 - t, b_t = b_0 - t$. Verificamos a mano para $t = 0$ que $a_t = a_0 - 0 = a_0$ y $b_t = b_0 - 0 = b_0$, lo cual es cierto porque aún no se ha ejecutado ninguna iteración del ciclo. Supondremos que la invariante se sigue cumpliendo para alguna t. Si la t-ésima evaluación de la condición fue falsa, entonces sabemos que $b_t = 0$ y siguiendo la invariante $b_t = b_0 - t \implies t = b_0$ y $a_t = a_0 - t = a_0 - b_0$. Como el ciclo termina y el valor que se devuelve es $a_t = a_0 - b_0 = a - b$, el resultado es correcto. Si la t-ésima evaluación de la condición fue verdadera entonces se ejecuta el ciclo y se evaluará la condición por $(t + 1)$-ésima vez. Durante el ciclo, las variables pasan a valer $a_{t+1} = a_t - 1, b_{t+1} = b_t - 1$ por las restas que ejecuta el algoritmo.

Reescribiendo obtenemos $a_{t+1} = a_t - 1 = (a_0 - t) - 1 = a_0 - (t+1)$ y $b_{t+1} = a_t - 1 = (b_0 - t) - 1 = b_0 - (t+1)$, por lo que la invariante se sigue cumpliendo.

- Demuestra que $n + 10 \in O(n)$.
  Usaremos $n_0 = 2, c = 10$ para demostrar que $n + 10 \leq 10n$ para $n \geq 2$. Verificaremos a mano para $n = 2$ y tenemos que $2 + 10 \leq 10(2)$, lo cual es correcto. Supondremos que es cierto para cierta n y verificamos por inducción para $n+1$. Evaluando para $n+1$ tenemos $(n+1)+10 \leq 10(n+1)$ y reescribimos para obtener $(n+10)+1 \leq (10n)+10$. Como $n+10 \leq 10n$ por la hipótesis de inducción y $1 \leq 10$, entonces la desigualdad es correcta.

- Demuestra que $5n \log_2 n \in \Omega(n)$.
  Usaremos $n_0 = 2, c = 5$ para demostrar que $5n \log_2 n \geq 5n$ para $n \geq 2$. Como la única diferencia entre ambos lados de la desigualdad es que el lado izquierdo tiene un factor $\log_2 n$ que es positivo para $n \geq 2$, la desigualdad es correcta.