Enunciado:

Resuelva los siguientes ejercicios sobre análisis de algoritmos, crecimiento de funciones, mejor y peor caso, caso promedio y reglas para calcular tiempos de ejecución.

1. Suponga que $T_1(N) = O(f(N))$ y $T_2(N) = O(f(N))$. Indique cuales de los siguientes enunciados son verdaderos.

   a) $T_1(N) + T_2(N) = O(f(N))$
   b) $T_1(N) - T_2(N) = o(f(N))$
   c) $T_1(N)/T_2(N) = O(1)$
   d) $T_1(N) = O(T_2(N))$

2. Un algoritmo ejecuta en 0,5 ms para un conjunto de datos de entrada de tamaño 100. ¿Qué tan grande son los datos de entrada si el algoritmo ejecuta durante 1 minuto, si su tiempo de ejecución es alguno de las siguientes opciones?

   a) $O(N \log N)$
   b) Cuadrático
   c) Cúbico

3. El análisis del tiempo computacional, $T(N)$, de un algoritmo arroja el resultado $T(N) = a \log N + bN$, donde a y b son constantes independientes de N. Discuta de qué factores pueden depender estos valores.

4. Usando la definición del conjunto de funciones $O(f(N))$, determine si las siguientes proposiciones son falsas o verdaderas. Justifique claramente sus repuestas y describa detalladamente cada paso de la demostración.

   a) $2^{2n} = O(2^n)$
   b) $2^{n+1} = O(2^n)$

5. Para cada uno de los siguientes fragmentos de código haga un análisis del tiempo de ejecución en términos de notación $O(f(n))$, $\Omega(g(n))$ y $\Theta(h(n))$.

   a)
   ```
   for(i = 0; i < n; ++i)
     for(j = 0; j < n * n; ++j)
       ++sum;
   ```

   b)
   ```
   for(i = 0; i < n; ++i)
     for(j = 0; j < i; ++j)
       ++sum;
   ```

   c)
   ```
   for(i = 0; i < n; ++i)
     for(j = 0; j < i * i; ++j)
       for(k = 0; k < j; ++k)
         ++sum;
   ```

d)
```
for(i = 1; i < n; ++i)
  for(j = 1; j < i * i; ++j)
    if(j % i == 0)
      for(k = 0; k < j; ++k)
        ++sum;
```

6. ¿Cuál es el tiempo de ejecución, $T(N)$, del siguiente ciclo como una función de N? Una vez calculado $T(N)$, demuestre a cuáles de los conjuntos de funciones $\Theta(f(N))$, $\Omega(g(N))$ o $O(h(N))$ pertenece. Justifique claramente su respuesta. Suponga que se satisface que $N \geqslant 6$.

```
for(int i = 5; i < N; i *= 2)
  sum += i * i;
```

7. Considere un arreglo de N números enteros. Diseñe un algoritmo que mueve todos los ceros, 0, al final del arreglo, manteniendo el orden de los elementos distintos de cero. La complejidad computacional del algoritmo debe ser $O(N)$ en tiempo de ejecución y $O(1)$ en almacenamiento en memoria, adicional al arreglo original.

   NOTA: Este problema puede resolverse recorriendo el arreglo una o dos veces.