Nombre: Guillermo Montemayor Marroquín A01722402

Grupo: 607

Quiz 1

1. **¿Qué es un algoritmo?**

Una serie de instrucciones para resolver un problema específico y obtener un resultado esperado. Por ejemplo, una receta de cocina, ordenar una lista, hacer que un robot siga un camino, resolver un laberinto, etc.

2. **Escriba la descripción matemática formal de la notación Θ. Además, explique con sus propias palabras qué significa esta notación. Puede apoyar su respuesta con una figura.**

   Definición matemática (de acuerdo a Cormen et al. [ver recursos]):

$$\Theta(g(n)) = \{f(n) : \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\}.$$

```
\begin{align*}
\Theta(g(n)) = \{f(n) : \text{there exist positive constants } c_1,
c_2, \text{ and } n_0 &\text{ such that }  \\
0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) &\text{ for all } n \geq n_0
\}.
\end{align*}
```

Θ limita los costos computacionales del algoritmo con c_1 (menos costoso) y c_2 (más costoso) para que f(n) esté en el medio de aquellos dos. En pocas palabras Θ(g(n)) representa el caso promedio.

3. **Escriba la descripción matemática formal de la notación O. Además, explique con sus propias palabras qué significa esta notación. Puede apoyar su respuesta con una figura.**

   Definición matemática (de acuerdo a Cormen et al. [ver recursos]):

$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}.$$

```
\begin{align*}
O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{
and } n_0 \text{ such that } 0 \leq f(n) \leq c g(n) \text{ for all }
n \geq n_0\}.
\end{align*}
```

El costo computacional de O(g(n)) es el máximo posible. También conocido como el peor caso, por ejemplo en bubble sort sería $O(n^2)$ para una lista ordenada de manera inversa.

4. **Escriba la descripción matemática formal de la notación Ω. Además, explique con sus propias palabras qué significa esta notación. Puede apoyar su respuesta con una figura.**

   Definición matemática (de acuerdo a Cormen et al. [ver recursos]):

$$\Omega(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}.$$

```
\begin{align*}
\Omega(g(n)) = \{f(n) : \text{there exist positive constants } c
\text{ and } n_0 \text{ such that } 0 \leq c g(n) \leq f(n) \text{
for all } n \geq n_0\}.
\end{align*}
```

El costo computacional de Ω(g(n)) es el menor posible. También conocido como el mejor caso, por ejemplo en bubble sort sería Ω(n) para una lista ya ordenada.

5. Realice el análisis de complejidad para el peor caso del algoritmo Bubble sort que se muestra en la Figura 1. Entregue su respuesta usando notación asintótica. El análisis de complejidad debe ser hecho como el visto en clase.

Figura 1: Algoritmo bubble sort que ordena un arreglo A = [a1, a2, . . . , an] de números. A.length retorna la longitud del arreglo y A[i] retorna el i-´esimo elemento del arreglo. El arreglo de indexa de 1 a n, donde n es el tamaño del arreglo.

```
BUBBLESORT(A)
1  for i = 1 to A.length − 1
2      for j = A.length downto i + 1
3          if A[j] < A[j − 1]
4              exchange A[j] with A[j − 1]
```

Instrucción	Costo (C)	Repeticiones
for i = 1 to A.length - 1	C_1	n
for j = A.length downto i+1	C_2	$\sum_{j=2}^{n} t_j$
if A[j] < A [j - 1]	C_3	$\sum_{j=2}^{n} (t_j - 1)$
exchange A[j] with A[j-1]	C_4	$\sum_{j=2}^{n} (t_j - 1)$

$$T(n) = C_1 n + C_2(\sum_{j=2}^{n} t_j) + C_3(\sum_{j=2}^{n} t_j - 1) + C_4(\sum_{j=2}^{n} t_j - 1)$$

$$T(n) = C_1 n + C_2(\frac{n(n-1)}{2}) + C_3(\frac{n(n-1)}{2} - 1) + C_4(\frac{n(n-1)}{2} - 1)$$

$$T(n) = n + \frac{n^2-n}{2} + \frac{n^2-n}{2} - 1 + \frac{n^2-n}{2} - 1$$

$$T(n) = n + 3(\frac{n^2-n}{2}) - 2$$

$$T(n) = \frac{3n^2-3n+2n-4}{2}$$

$$T(n) = \frac{3n^2-n-4}{2} \approx T(n) = n^2 \rightarrow O(n) = n^2$$

Peor caso: $O(n^2)$
Mejor caso: $\Omega(n)$

Recursos:

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, Third Edition* (3rd ed.) (p.43-53). The MIT Press.