Tarea 6: Complejidad y Algoritmia

I. Preguntas

1. Defina los siguientes conceptos:

a) Algoritmo **b)** Eficiencia **c)** Complejidad de un programa
d) Espacio de memoria **e)** Tiempo de ejecución

2. ¿Para que se emplea el **análisis asintótico**?

3. Defina las siguientes notaciones asintóticas:

a) O **b) W** **c) Q** **d) o** **e) w**

4. Compruebe lo siguiente:

- Determinar si $\mathbf{10^6}$ es **O** $\mathbf{(n^2)}$**.**
- Determinar si **n(n - 1) / 2** es **O** $\mathbf{(n^2)}$**.**
- Determinar si $\mathbf{n^2}$ es **O (n).**
- Determinar si **n / 2** es **O** $\mathbf{(n^2)}$**.**
- Determinar si $\mathbf{10^6}$ es **W** $\mathbf{(n^2)}$**.**
- Determinar si **n(n - 1) / 2** es **W** $\mathbf{(n^2)}$**.**
- Determinar si $\mathbf{n^2}$ es **W (n).**
- Determinar si **n / 2** es **W** $\mathbf{(n^2)}$**.**
- Determinar si $\mathbf{10^6}$ es **Q** $\mathbf{(n^2)}$**.**
- Determinar si **n(n - 1) / 2** es **Q** $\mathbf{(n^2)}$**.**
- Determinar si $\mathbf{n^2}$ es **Q (n).**
- Determinar si **n / 2** es **Q** $\mathbf{(n^2)}$**.**

5. Enumere las **ventajas de la notación asintótica**.

6. Describa la clasificación de **algoritmos tratables e intratables**.

7. Cuando se comprueba que un **algoritmo es intratable** ¿Qué es posible hacer?

8. Describa brevemente los **casos** que se presentan en un algoritmo.

9. Describa las **reglas generales** a seguir para el análisis de casos.

10. Para los siguientes algoritmos, describa la forma de obtener su **complejidad** describiendo (según el caso) cuando se presenta el peor caso, el caso promedio y el mejor caso.

- Ordenamiento por inserción.
- Torres de Hanoi.
- Búsqueda Binaria.

11. Describa y dé un ejemplo de cada una de las técnicas algorítmicas.

12. Compruebe la complejidad de los siguientes algoritmos (vistos en clase):

a) Secuencia **Fibonacci** Recursiva: **Q(Æ^n)**

Donde el número de llamadas iterativas para `Fibonacci(n)` es:

Llamadas = (2 / Ō5) * (Æ ^{n} + (1 / Æ)^{n}) - 2

Donde: **Æ = (1 + Ō5) / 2**; es la llamada **razón áurea**.

b) Fractal de **Hilbert**: Llamadas: **$4^{n} - 1$** **Q(4^{n})** con **n** = orden

c) Fractal **Sierpinski**: Llamadas: **4^{n+1}** **Q(4^{n})** con **n** = orden