

Facultad Politécnica – Ingeniería Informática  
**Algoritmos y Estructura de Datos III – 2025 – 2do. Periodo – Sección TQ/TR**  
**Tarea 2-U2**

**Ejercicios teóricos (25p).** Debe incluirse en un archivo PDF que puede ser un escaneado de soluciones hechas a mano.

1. Encuentre el costo asintótico en términos de  $O$  (o grande) de la siguiente recurrencia considerando que  $T(n) = 1$  si  $n = 0$ . Puede utilizar cualquier método conocido para encontrarlo (4p)

$$T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log^2 n} \text{ si } n > 0$$

2. Encuentre el orden de crecimiento en término de  $O$  ( $O(g(n))$ ) de las siguientes sumas (utilice la función  $g(n)$  más simple y precisa posible). Justificar (2p cada uno).

a.  $\sum_{i=1}^{\log_2 n} 4^i$

b.  $\sum_{i=1}^n (i+1) 3^{i-1}$

c.  $\sum_{i=2}^{n-1} \log_2 i^2$

d.  $\sum_{i=1}^n \sum_{j=1}^{i-1} \sum_{k=1}^{j-1} (i+j+k)$

3. Resuelva las siguientes recurrencias, por el método que usted conozca. Justificar las respuestas. Asumir que  $T(n)$  es constante para una  $n$  suficientemente pequeña, por ejemplo  $T(n)=C$  (constante) si  $n=1$ . (2p. cada uno)

a.  $T(n) = T(9n/10) + n^2$

b.  $T(n) = T(n-2) + \log_2 n^2$

c.  $T(n) = 3T(n-2) + \sqrt{n}$

d.  $T(n) = \sqrt{n} T(\sqrt{n}) + n$

4. Encuentre la forma cerrada de la siguiente expresión (3p) y luego demuestre por inducción matemática la correctitud de dicha forma cerrada (2p).

$$\sum_{i=1}^n i^2$$

## Ejercicio Práctico (20 puntos)

Genere vectores de tamaño  $N$  de número enteros aleatorios, donde  $N$  comienza en 50.000 y crece hasta 1.000.000, con incrementos de 50.000. El dominio de los valores de cada elemento va de 50.000 a 1.000.000. Luego realice  $N$  búsquedas aleatorias utilizando la estrategia de **búsqueda binaria** (esto es, busque un elemento aleatorio en el propio arreglo) y  $N$  búsquedas aleatorias usando una **búsqueda lineal**. El valor buscado será entre los 1 y  $N$  seleccionado en forma aleatoria (genere la posición a elegir en forma aleatoria).

Cada función de búsqueda debe hacerse en un método que debe ser definido como genérico y que soporte tipos comparables. Genere la tabla parecida a la que se muestra abajo:

N	Búsqueda binaria				Búsqueda Lineal			
	T(n) en ms	T/N	T/(N log <sub>2</sub> N)	T/N <sup>2</sup>	T(n) en ms	T/N	T/(N log <sub>2</sub> N)	T/N <sup>2</sup>
50000								
100000								
150000								
..								

### Observaciones:

- Para aplicar la búsqueda binaria el vector debe estar ordenado, para ello implemente el algoritmo de **TimSort**. Para ello crea una clase separada denominada *Util*, donde deberá estar el código del algoritmo de ordenación. *El cálculo de tiempo no debe incluir la ordenación*. El método debe ser genérico para los tipos adecuados de ordenación.
- Su programa debe generar la tabla mencionada, es decir no debe solo transcribir los resultados en un documento. Utilice un formato legible para los decimales, utilice *printf()* para ello con las cadenas de formato adecuadas.
- El tiempo de ejecución debe estar en milisegundos con la función que fue indicada en la clase en laboratorio.

Al final de su implementación coloque, en forma de comentario en el propio código, los resultados que obtuvo en sus experimentos, es decir la tabla misma con los resultados. Luego explique los valores obtenidos y el porqué de las tendencias de cada columna (que corresponden a la Búsqueda Binaria y la Búsqueda Lineal). Además indicar que entorno de que utilizó para las pruebas (Hardware (CPU,RAM), Sistema Operativo, versión de Java, etc.). (2p)

Rúbrica	Puntaje
Implementación correcta de funciones de búsqueda definidos como Generic	4
Implementación del Algoritmo de ordenación en clase separada denominada Util	2
Implementación correcta del experimento con generar la generación de la tabla de acuerdo al modelo mostrado	10
Comentarios relevantes sobre los hallazgos en el experimento. Se debe de incluir todo lo solicitado.	4