

Nombre y Apellido: \_\_\_\_\_ Nro. CIC: \_\_\_\_\_

**Tema 1 (10p)**

Considere las siguientes funciones escritas en Java

<pre>int descubremel ( int [] a ) {     int n = a.length;     int tmp = 0, i, j;     for ( i=0; i &lt; n ; i++ )         for ( j=i+1; j &lt; n ; j++ )             if ( Math.abs(a[j]-a[i]) &gt; tmp )                 tmp = Math.abs(a[j]-a[i]);     return tmp; }</pre>	<pre>void descubreme2 ( int [] a ) {     int n = a.length;     int tmp, i, j;     for ( i=0; i &lt; n ; i++ )         if ( a[i]%2 == 0 )         {             tmp=a[i];             for ( j=i; j &gt; 0 ; j-- )                 a[j] = a[j-1];             a[0]=tmp;         } }</pre>
---	---

Por cada una de ellas, se pide:

- ¿Qué calcula la función? (2p)
- Calcule teniendo en cuenta el peor caso: T(n) y O. (4p)
- Según a), proponga una función con mejor rendimiento, lo que tiene que fundamentar (4p)

**Tema 2 (12p)**

Utilizando las definiciones de  $O$  y  $\Omega$ , encontrar la  $n_0$  y  $c$  para las siguientes funciones (utilice el nombre de las constantes indicadas en cada caso).

$f(n)$	$O$		$\Omega$	
	$n_0$	$c$	$n_0$	$c$
$c_1 n$				
$c_2 n^3 + c_3$				
$c_6 2^n + c_7 n^6$				

**Tema 3 (10p)**

Complete las siguientes oraciones de manera corta y precisa:

- El lazo invariante consiste en \_\_\_\_\_
- El factor de carga en una tabla de dispersión con resolución de colisión abierta indica \_\_\_\_\_
- Una excepción derivada de *RuntimeException* tiene la característica principal consistente en \_\_\_\_\_
- Un algoritmo es eficiente cuando \_\_\_\_\_
- Un iterador en una estructura de datos consiste en \_\_\_\_\_

Defina un iterador utilizando el API de Java para la estructura de datos Pila genérica (asuma que ya se tiene implementada las operaciones básicas sobre esta estructura: *pop()*, *push()*, *empty()*, etc.)

#### Tema 4 (8p)

Coloque la letra de la respuesta correcta al lado de la frase (existe solo una) (-1p por cada respuesta marcada e incorrecta)

1. \_\_\_\_ Un algoritmo toma 10 segundos para un tamaño N de 50. Si el algoritmo es cúbico, aproximadamente cuanto tiempo tomará para resolver un problema de tamaño 100.  
a) 40 segundos      b) 8000 segundos      c) 2000 segundos      d) 80 segundos      e) ninguna de las anteriores
2. \_\_\_\_ Un algoritmo toma 6 segundos para un tamaño de 100 y 10 minutos para resolver el un problema de tamaño 1000. ¿Cuál es el comportamiento asintótico del tiempo de ejecución del algoritmo?  
a) constante      b) lineal      c) cuadrático      d) cúbico      e) ninguna de las anteriores
3. \_\_\_\_ En Java, se puede invocar a un método estático en los siguientes casos.  
a) Solamente con el nombre de la clase que lo contiene      b) Solamente con la instancia de la clase  
c) Con el nombre o con la instancia de la clase      d) Con el nombre de la clase Padre      e) Solo con la clase Object
4. \_\_\_\_ Para la inserción de un solo ítem en un árbol AVL de  $n$  elementos, ¿el máximo número de rotaciones requerida es? (asuma que una rotación doble es una sola rotación).  
a) 1      b) 2      c) aprox.  $\log n$       d) aprox.  $1.44 \log n$       e) ninguna de las anteriores
5. \_\_\_\_ Los siguientes ítems son insertados en un árbol AVL: 1,2,3,8,6. ¿Cuántas rotaciones son realizadas?  
a) no hay rotaciones      b) una simple      c) una doble      d) una simple y una doble      e) ninguna de las anteriores
6. \_\_\_\_ Dada la rutina *recurs* :  

```
int recurs ( int n ) {  
    if ( n == 0 )  
        return 0;  
    else  
        return n + recurs ( n/2 ) + recurs ( n/2 + 1 );  
}
```

la misma viola las siguientes reglas de la recursión:  
a) No tiene caso base      b) No progresa      c) Realiza trabajo redundante      d) b y c      e) a, b y c
7. \_\_\_\_ El algoritmo de *Strassen* para multiplicación de dos matrices de tamaño  $N \times N$ , realiza siete(7) llamadas recursivas para multiplicar dos matrices de  $N/2 \times N/2$ . El tiempo adicional que se suma a las llamadas recursivas es cuadrático. Entonces, la solución se encuentra en:  
a)  $O(N^2)$       b)  $O(N^3)$       c)  $O(N^{2.81})$       d)  $O(N^{2.5})$       e) ninguna de las anteriores.
8. \_\_\_\_ ¿Cual de los siguientes costos son iguales en una tabla Hash?  
a) Inserción y búsqueda exitosa      b) Inserción y búsqueda sin éxito      c) Búsqueda con éxito y búsqueda sin éxito  
d) Inserción, búsqueda exitosa y búsqueda sin éxito      e) ninguna de las anteriores.

#### Tema 5 (10p)

- a) Encuentre el costo asintótico en términos de  $\Theta$  de la siguiente recurrencia. Puede utilizar cualquier método conocido para encontrarlo (5p)

$$T(n) = \begin{cases} 2T(n-1) + 1 & \text{si } n > 0 \\ 0 & \text{si } n = 0 \end{cases}$$

- b) Encuentre la  $O$  más ajustada para esta sumatoria y luego dé un código posible en Java que el corresponda (5p)

$$\sum_{i=1}^{n-1} (n-i)i$$

#### Tema 6 (10p)

Encuentre la forma cerrada de la sumatoria de los  $n$  primeros números múltiplos de 5. Luego demuestre por inducción matemática que la forma cerrada es correcta.

Por ejemplo si  $n=3$ , la suma sería  $5+10+15 = 30$