Pseudocódigo Avanzado: Recursión y Análisis de Complejidad

Nombre: _____

Fecha: _____

Puntaje: _____

1.

¿Cuál es la característica esencial de una función recursiva?

1. Llamarse a sí misma
2. Utilizar un bucle
3. Devolver un valor entero

2.

En una función recursiva, ¿qué condición evita que ocurra un bucle infinito?

1. Caso base
2. Caso recursivo
3. Parámetro por valor
4. Retorno explícito

3.

¿Qué estructura de datos utiliza implícitamente el mecanismo de recursión para gestionar las llamadas pendientes?

1. Pila (stack)
2. Cola (queue)
3. Lista enlazada
4. Árbol binario

4.

En la función recursiva factorial, el caso base para n=0 retorna el valor ___.

Respuesta: _____

5.

¿Qué tipo de recursión se presenta cuando una función se llama a sí misma más de una vez en el mismo paso recursivo?

1. Recursión lineal
2. Recursión en árbol
3. Recursión de cola
4. Recursión indirecta

6.

Complejidad temporal de la función recursiva factorial (sin optimización) para entrada n:

1. O(n)
2. O(n^2)
3. O(2^n)

7.

Dada la recurrencia T(n) = T(n-1) + c, con T(0)=c, ¿cuál es su complejidad asintótica?

1. O(1)
2. O(log n)
3. O(n^2)
4. O(n)

8.

Para la secuencia de Fibonacci recursiva clásica, su complejidad temporal es aproximadamente:

1. O(n)
2. O(n^2)
3. O(log n)
4. O(2^n)

9.

La complejidad espacial de la función recursiva factorial (sin optimización) es O(___). (Expresa con la notación asintótica, ej: n, n^2, log n)

Respuesta: _____

10.

¿Cuál de las siguientes estrategias permite reducir la complejidad temporal de Fibonacci recursivo de exponencial a lineal?

1. Recursión de cola
2. Recursión anidada
3. Memoización
4. División y vencerás

11.

En recursión de cola, la llamada recursiva es la última operación. ¿Qué ventaja principal ofrece?

1. Menor uso de memoria
2. Mayor legibilidad
3. Evita el caso base

12.

Analiza el siguiente pseudocódigo recursivo: funcion misterio(n): si n=0: retornar 0; sino retornar n + misterio(n-1). ¿Qué calcula?

1. n!
2. n^2
3. Suma 1..n
4. 2^n

13.

En la función recursiva suma_digitos(n), el caso base ocurre cuando n < 10, y se retorna ___.

Respuesta: _____

14.

¿Cuál de los siguientes algoritmos típicamente NO se implementa de forma recursiva?

1. Recorrido de árbol binario
2. Búsqueda binaria
3. Euclides
4. Ordenamiento por burbuja

15.

La complejidad temporal de la función de Ackermann (A(m,n)) es:

1. O(m*n)
2. O(2^(m+n))
3. Extremadamente rápida (superexponencial)
4. O(m log n)

16.

En el análisis de complejidad, notación Theta (Θ) se usa para:

1. Cota superior asintótica
2. Cota inferior asintótica
3. Cota ajustada asintótica

17.

¿Cuál es la complejidad temporal del siguiente algoritmo recursivo? funcion f(n): si n<=1: retornar 1; sino retornar f(n/2) + f(n/2). (Asumiendo que la suma es O(1))

1. O(log n)
2. O(n log n)
3. O(n)
4. O(2^n)

18.

La recursión indirecta ocurre cuando:

1. Una función se llama a sí misma
2. Dos o más funciones se llaman mutuamente
3. La llamada recursiva no es la última instrucción
4. Se usa una pila explícita

19.

Para el algoritmo recursivo que calcula la potencia x^n con la estrategia de exponenciación rápida, su complejidad temporal es O(___).

Respuesta: _____

20.

En el análisis de la complejidad espacial de la recursión, ¿cuál de los siguientes factores determina principalmente el espacio ocupado en la pila?

1. Número de llamadas recursivas simultáneas
2. Tamaño de los parámetros
3. Número de variables locales
4. Tipo de recursión (lineal o árbol)