1. ¿Cuál de las siguientes no es una característica de las rutinas recursivas?

a. Se llaman a sí mismas.

b. Cada llamada realiza su trabajo en una versión más pequeña del mismo

problema.

c. Cuando una versión más pequeña del problema es demasiado compleja, el control

pasa de nuevo al que llama para que intente un enfoque diferente.

d. Algunas versiones del problema no requieren llamar a la rutina recursiva.

1. Si un programa llama a triangular(100) utilizando la definición del Listado 6-1,

¿cuál es el número máximo de “copias” de la función triangular()

en ejecución que existen en un momento dado?

1. ¿Dónde se almacenan las copias del argumento pasado a la función triangular(), mencionado en la pregunta 2?

a. en una variable de la función triangular()

b. en un campo de la clase Triangular

c. en una variable de marcador de posición del registro de descripción del problema

d. en una pila

1. Suponga una llamada a triangular(100) como en la pregunta 2. ¿Cuál es el valor

de nth justo después de que la función triangular() devuelva por primera vez un valor

de caso no base?

1. Verdadero o falso: En la función triangular() del Listado 6-1, los valores de retorno se almacenan en la pila.
2. En la función anagrams(), a una cierta profundidad de recursión, suponga que una

versión de la función está trabajando con la cadena "que". Cuando este

método llama a una nueva versión de sí mismo, ¿con qué letras estará

trabajando la nueva versión?

1. En la sección “Una búsqueda binaria recursiva”, se comparó la forma original basada en bucles con una forma recursiva del método find(). ¿Cuál

de las siguientes afirmaciones no es verdadera?

a. El rango de búsqueda comienza con toda la matriz, y solo la versión recursiva puede

funcionar en un subrango pasado a través de argumentos.

b. Ambas formas del programa dividen el rango de búsqueda repetidamente por la mitad.

c. Si no se encuentra la clave, la versión de bucle retorna cuando se cruzan los

límites del rango, pero la versión recursiva finaliza cuando la profundidad

recursiva es mayor que la mitad del rango de búsqueda inicial.

d. Si se encuentra la clave, la versión de bucle retorna desde el método completo,

mientras que la versión recursiva retorna desde un nivel de recursión.

1. ¿Qué tipo de subproblema se resuelve en las llamadas recursivas del método TowerOfHanoi.solve() (Listado 6-3) en comparación con el

problema general?

1. El algoritmo en el método TowerOfHanoi.solve() implica

a. dividir la cantidad de discos en el husillo de origen a la mitad.

b. cambiar qué husillos son el origen y el destino.

c. quitar los discos pequeños de todos los husillos para mover los discos grandes.

d. mover un disco pequeño y luego una pila de discos más grandes.

1. ¿Cuál de las siguientes afirmaciones no es cierta sobre el método Mergesort.merge() en el Listado 6-

5?

a. Su algoritmo puede manejar matrices de diferentes tamaños.

b. Debe buscar en la matriz de destino para encontrar dónde colocar el siguiente elemento.

c. No es recursivo.

d. Copia continuamente el elemento más pequeño independientemente de en qué matriz de

entrada se encuentre.

1. La desventaja de mergesort es que

a. no es recursivo.

b. utiliza más memoria.

c. aunque es más rápido que el ordenamiento por inserción, es mucho más lento que el

ordenamiento por burbuja.

d. es complicado de implementar.

1. Al utilizar la versión recursiva de mergesort() en el Listado 6-4, ¿qué

profundidad recursiva se alcanzará en una llamada para ordenar una matriz de 1024 celdas?

1. Además de un bucle, a menudo se puede utilizar un \_\_\_\_\_\_\_\_\_\_\_ en lugar de

la recursión.

1. En el procedimiento descrito para convertir una función recursiva en un

algoritmo no recursivo, ¿cómo se almacenan los argumentos de la función y las variables locales

de la versión recursiva?

1. En el procedimiento descrito para convertir una función recursiva en una

función no recursiva, ¿qué prueba se utiliza para decidir cuándo debe retornar la función?