

# Cuestionario Global de bittercios

🕒 Fecha de Creación	@20 de diciembre de 2023 15:12
👤 Asignatura	FAL
🕒 Fecha de Modificación	@20 de diciembre de 2023 15:15

## Pregunta 1

Correcta

Se puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

La función de tiempo  $T(n)$  de un algoritmo  $A$  satisface la siguiente recurrencia:

$$T(0) = 1$$

$$T(n) = n + 2T(n/2), n > 0$$

¿Cuáles de los siguientes órdenes caracterizan de manera más precisa la complejidad de este algoritmo?

Seleccione una o más de una:

- ☐ a.  $O(\log n)$
- ☐ b.  $O(1)$
- ☐ c.  $O(n)$
- ☒ d.  $O(n \log n)$  ✓
- ☐ e.  $O(\sqrt{n})$

**Cierto.** La recurrencia se ajusta al patrón de división, con  $a = 2$ ,  $k = 1$ , y  $b = 2$ . Por tanto,  $a = 2 = b^k = 2^1 = 2$ . Como resultado,  $T(n) \in O(n^k \log n) = O(n^1 \log n) = O(n \log n)$ .

- a. **Falso.** Se obtiene el orden de complejidad  $O(n \log n)$
- b. **Falso.** Se obtiene el orden de complejidad  $O(n \log n)$
- c. **Falso.** Se obtiene el orden de complejidad  $O(n \log n)$
- d. **Cierto.** La recurrencia se ajusta al patrón de división, con  $a = 2$ ,  $k = 1$ , y  $b = 2$ . Por tanto,  $a = 2 = b^k = 2^1 = 2$ . Como resultado,  $T(n) \in O(n^k \log n) = O(n^1 \log n) = O(n \log n)$ .
- e. **Falso.** Se obtiene el orden de complejidad  $O(n \log n)$

La respuesta correcta es:  $O(n \log n)$

## Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

¿Cuáles de los siguientes elementos intervienen en la demostración de la **corrección parcial** de un algoritmo iterativo?

Seleccione una o más de una:

- ☒ a. La postcondición. ✓ **Cierto.** Debe asegurarse que los estados finales siempre verifican la postcondición.
- ☐ b. La expresión de cota.
- ☒ c. El invariante. ✓ **Cierto.** Caracteriza qué estados son ciertos inmediatamente antes de comenzar la iteración, y justo al final de cada iteración.
- ☒ d. La precondición. ✓ **Cierto.** Únicamente deben tenerse en cuenta los estados que verifican la precondición.
- ☐ e. La complejidad asintótica.

- a. **Cierto.** Debe asegurarse que los estados finales siempre verifican la postcondición.
- b. **Falso.** Se utiliza para demostrar que el algoritmo termina (la corrección parcial no se ocupa de la terminación).
- c. **Cierto.** Caracteriza qué estados son ciertos inmediatamente antes de comenzar la iteración, y justo al final de cada iteración.
- d. **Cierto.** Únicamente deben tenerse en cuenta los estados que verifican la precondición.
- e. **Falso.** Tiene que ver con la eficiencia del algoritmo, pero no se usa en la demostración de su terminación.

Las respuestas correctas son: La postcondición, El invariante, La precondición.

### Pregunta 3

Correcta

Se puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

¿Cuáles de los siguientes predicados podremos afirmar que se cumplen *si y sólo si* en las  $n$  primeras posiciones del array  $a$  se encuentra almacenado un vector de enteros, *todos ellos pares*?

Seleccione una o más de una:

- ☒ a.  $\forall i : 0 \leq i < n : a[i] \% 2 = 0$  ✓ **Cierto.** El predicado dice, precisamente, que **todos** los elementos deben ser pares.
- ☒ b.  $\neg \exists i : 0 \leq i < n : a[i] \% 2 = 1$  ✓ **Cierto.** El predicado dice que no es cierto que existan elementos impares en el vector. Por tanto, todos los elementos en el vector serán pares.
- ☐ c.  $\exists i : 0 \leq i < n : a[i] \% 2 = 0$
- ☐ d.  $\neg \forall i : 0 \leq i < n : a[i] \% 2 = 1$
- ☐ e.  $(\# i : 0 \leq i < n : a[i] \% 2 = 0) > 0$

- a. **Cierto.** El predicado dice, precisamente, que **todos** los elementos deben ser pares.
- b. **Cierto.** El predicado dice que no es cierto que existan elementos impares en el vector. Por tanto, todos los elementos en el vector serán pares.
- c. **Falso.** El predicado dice que puede haber elementos pares, pero no dice que todos ellos lo sean.
- d. **Falso.** El predicado dice que no es cierto que todos los elementos sean impares, pero ello no quiere decir que no pueda haber impares en el vector.
- e. **Falso.** El predicado dice que en el vector hay, al menos, un elemento par. Pero esto no quiere decir que todos los elementos sean pares.

Las respuestas correctas son:  $\forall i : 0 \leq i < n : a[i] \% 2 = 0$   
,  $\neg \exists i : 0 \leq i < n : a[i] \% 2 = 1$

### Pregunta 4

Parcialmente correcta

Se puntúa 0,67 sobre 1,00

🚩 Desmarcar

Considérese el siguiente algoritmo:

P:  $0 \leq n \leq \text{tam}(a)$

```
int algo(int a[], int n) {  
    int resul = 1;  
    int i=n-1;  
    while (i >= 0) {  
        resul = resul * a[i];  
        i--;  
    }  
    return resul;  
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- ☒ a.  $i$  es una expresión de cota del bucle. ✓ **Cierto.**  $i$  nunca será menor que  $-1$  (por tanto, está limitado inferiormente), y, además, decrece en cada iteración.
- ☐ b.  $n - i$  es una expresión de cota del bucle.
- ☒ c. La postcondición puede especificarse como  $\text{resul} = \prod i : 0 \leq i < n : a[i]$ . ✓ **Cierto.** El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.
- ☐ d.  $\text{resul} = \prod u : 0 \leq u < i : a[u]$  es un invariante del bucle.
- ☐ e.  $\text{resul} = \prod u : i < u < n : a[u]$  es un invariante del bucle.

- a. **Cierto.**  $i$  nunca será menor que  $-1$  (por tanto, está limitado inferiormente), y, además, decrece en cada iteración.
- b. **Falso.**  $n - i$  crece en cada iteración.
- c. **Cierto.** El algoritmo calcula el producto de todos los elementos del vector, que es lo que especifica precisamente este predicado.
- d. **Falso.** El bucle va acumulando en  $\text{resul}$   $a[n-1], a[n-2] * a[n-1], a[n-3] * a[n-2] * a[n-1] \dots$ , mientras que este predicado lo que dice es que en  $\text{resul}$  se van acumulando  $a[0], a[0] * a[1], a[0] * a[1] * a[2] \dots$
- e. **Cierto.** El bucle va acumulando en  $\text{resul}$   $a[n-1], a[n-2] * a[n-1], a[n-3] * a[n-2] * a[n-1] \dots$ , que es lo que especifica este predicado (otra cosa es que sea suficientemente fuerte para demostrar la corrección parcial del algoritmo; a este respecto, falta también indicar el rango de variación de  $i$ , pero este hecho no quita que el predicado siga siendo un invariante).

Las respuestas correctas son:  $i$  es una expresión de cota del bucle.  
, La postcondición puede especificarse como  $\text{resul} = \prod i : 0 \leq i < n : a[i]$ .  
,  $\text{resul} = \prod u : i < u < n : a[u]$  es un invariante del bucle.

?

### Pregunta 5

Correcta

Se puntúa 1,00 sobre 1,00

🚩 Desmarcar

Sea  $t(n) = n \log_2 n + 5n + 2 \log_2 n + 190n^2 + 18$  la función de tiempo de un algoritmo  $A$ . El coste de  $A$  está en:

Seleccione una o más de una:

- ☐ a.  $\Theta(1)$
- ☐ b.  $\Theta(n)$
- ☐ c.  $\Theta(n \log n)$
- ☒ d.  $\Theta(n^2)$  ✓ **Cierto.** El término que crece más rápido es  $n^2$ .
- ☐ e.  $\Theta(\log n)$

- a. **Falso.** El término que crece más rápido es  $n^2$ .
- b. **Falso.** El término que crece más rápido es  $n^2$ .
- c. **Falso.** El término que crece más rápido es  $n^2$ .
- d. **Cierto.** El término que crece más rápido es  $n^2$ .
- e. **Falso.** El término que crece más rápido es  $n^2$ .

La respuesta correcta es:  $\Theta(n^2)$

**Pregunta 6**

Parcialmente correcta

Se puntúa 0,17 sobre 1,00

Desmarcar

Considérese la siguiente función recursiva:

```
int algo(int m, int n) {
    if (m == 0) {
        return n + 1;
    }
    else if (n == 0) {
        return algo(m - 1, 1);
    }
    else {
        return algo(m - 1, algo(m, n - 1)); /**/
    }
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- ☒ a. Contiene recursión múltiple. **Cierto.** En `/**/` se realizan dos llamadas recursivas para un mismo caso recursivo.
- ☐ b. En `/**/` hay una llamada recursiva final y otra no final.
- ☐ c. Es una función recursiva final.
- ☒ d. Las dos llamadas recursivas en `/**/` son no finales. **Falso.** La llamada externa es final.
- ☒ e. Es una función recursiva no final. **Cierto.** Contiene una llamada no final: la llamada interna en `/**/`.

- a. **Cierto.** En `/**/` se realizan dos llamadas recursivas para un mismo caso recursivo.
- b. **Cierto.** La llamada externa es final, la interna es no final, ya que, una vez finalizada, debe realizarse la llamada externa.
- c. **Falso.** Contiene una llamada no final: la llamada interna en `/**/`.
- d. **Falso.** La llamada externa es final.
- e. **Cierto.** Contiene una llamada no final: la llamada interna en `/**/`.

Las respuestas correctas son: Contiene recursión múltiple., En `/**/` hay una llamada recursiva final y otra no final., Es una función recursiva no final.**Pregunta 7**

Correcta

Se puntúa 1,00 sobre 1,00

Marcar pregunta

¿Cuál es el papel de las estrategias de poda en los algoritmos de "vuelta atrás"?:

Seleccione una o más de una:

- ☐ a. Medir el número de soluciones exploradas por el algoritmo.
- ☒ b. Mejorar la eficiencia del algoritmo. **Cierto.** Permite reducir el número de soluciones exploradas, por lo que mejora la eficiencia.
- ☒ c. Reducir la cantidad de soluciones a explorar. **Cierto.** Permiten determinar que, bien no se van a alcanzar más soluciones, bien no se va a poder mejorar la solución actual.
- ☐ d. Permitir decidir si una solución final es óptima o no.
- ☐ e. Priorizar qué soluciones parciales se exploran antes, y qué soluciones parciales se exploran después.

- a. **Falso.** No mide el número de soluciones exploradas, sino que permite reducir alternativas.
- b. **Cierto.** Permite reducir el número de soluciones exploradas, por lo que mejora la eficiencia.
- c. **Cierto.** Permiten determinar que, bien no se van a alcanzar más soluciones, bien no se va a poder mejorar la solución actual.
- d. **Falso.** Dicha decisión depende del criterio de coste, no de la estrategia de poda.
- e. **Falso.** Permiten decidir si continuar o no la exploración, no el orden en el que deben explorarse las soluciones.

Las respuestas correctas son: Mejorar la eficiencia del algoritmo., Reducir la cantidad de soluciones a explorar.

**Pregunta 8**

Correcta

Se puntúa 1,00 sobre 1,00

Marcar pregunta

Sea  $P(x)$  un predicado. ¿Cuáles de las siguientes afirmaciones son ciertas?:

Seleccione una o más de una:

- ☐ a.  $\Sigma x : false : P(x)$  vale 1.
- ☐ b.  $\Sigma x : false : P(x)$  vale true.
- ☐ c.  $\Pi x : false : P(x)$  vale 0.
- ☒ d.  $\Pi x : false : P(x)$  vale 1. **Cierto.** Por definición, el productorio de un conjunto vacío de objetos vale 1.
- ☒ e.  $\Sigma x : false : P(x)$  vale 0. **Cierto.** Por definición, el sumatorio de un conjunto vacío de objetos vale 0.

- a. **Falso.** Vale 0.
- b. **Falso.** Esta expresión no es un predicado, sino una expresión numérica (su valor, por tanto, nunca podrá ser un valor booleano).
- c. **Falso.** Vale 1.
- d. **Cierto.** Por definición, el productorio de un conjunto vacío de objetos vale 1.
- e. **Cierto.** Por definición, el sumatorio de un conjunto vacío de objetos vale 0.

Las respuestas correctas son:  $\Pi x : false : P(x)$  vale 1.,  $\Sigma x : false : P(x)$  vale 0.

**Pregunta 9**

Parcialmente  
correcta  
Se puntúa 0,67  
sobre 1,00

Desmarcar

¿Cuáles de los siguientes son algoritmos "divide y vencerás"?

Seleccione una o más de una:

- ☐ a. El método de ordenación de burbuja.
- ☒ b. El método de búsqueda binaria. ✓ **Cierto.** Descompone el problema en un subproblema con, aproximadamente, la mitad de tamaño que el problema original.
- ☐ c. El algoritmo de búsqueda de la salida en un laberinto.
- ☒ d. El método de ordenación por mezcla (*mergesort*). ✓ **Cierto.** Descompone el problema en dos subproblemas independientes de, aproximadamente, el mismo tamaño.
- ☐ e. El algoritmo de la transformada rápida de Fourier.

- a. **Falso.** Puede verse como un algoritmo que, tras un coste lineal, genera un nuevo subproblema de ordenación con tamaño únicamente una unidad menor.
- b. **Cierto.** Descompone el problema en un subproblema con, aproximadamente, la mitad de tamaño que el problema original.
- c. **Falso.** Es un algoritmo de "vuelta atrás".
- d. **Cierto.** Descompone el problema en dos subproblemas independientes de, aproximadamente, el mismo tamaño.
- e. **Cierto.** Reduce el cálculo de la transformada discreta de Fourier de una secuencia al cálculo de las transformadas de las subsecuencias en posiciones pares y en posiciones impares.

Las respuestas correctas son: El método de búsqueda binaria., El método de ordenación por mezcla (*mergesort*)., El algoritmo de la transformada rápida de Fourier.

**Pregunta 10**

Parcialmente  
correcta  
Se puntúa 0,17  
sobre 1,00

Desmarcar

Sabiendo que  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$ , ¿cuáles de las siguientes afirmaciones son verdaderas?:

Seleccione una o más de una:

- ☒ a.  $f(n) \in O(g(n))$  ✓ **Cierto.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- ☒ b.  $g(n) \in \Theta(f(n))$  ✗ **Falso.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- ☐ c.  $f(n) \in \Omega(g(n))$
- ☐ d.  $g(n) \in \Omega(f(n))$
- ☐ e.  $g(n) \in O(f(n))$

- a. **Cierto.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- b. **Falso.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- c. **Falso.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- d. **Cierto.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .
- e. **Falso.**  $g(n)$  crece asintóticamente más rápido que  $f(n)$ .

Las respuestas correctas son:  $f(n) \in O(g(n))$   
,  $g(n) \in \Omega(f(n))$