Examen de teoría
Algoritmia y optimización
Grado en Ingeniería en Inteligencia Artificial
Curso 2024-25

Modalidad 0

Instrucciones del examen

- Responde en la **plantilla para respuestas** proporcionada.
- Cada respuesta incorrecta resta $\frac{1}{3}$ de una respuesta correcta.
- El tiempo para la prueba es de **1 hora**.

Pregunta 1. Con respecto a la complejidad del siguiente algoritmo.

```
def producto_recursivo(v):
    if len(v) == 1:
        return v[0]
    m = len(v) // 2
    i = producto_recursivo(v[:m])
    d = producto_recursivo(v[m:])
    return i * d
```

Asumiendo que las operaciones elementales tienen coste 1:

A. Relación $T(n) = 2T(\frac{n}{2}) + \mathcal{O}(1)$, y por tanto, $T(n) \in \mathcal{O}(n)$
B. Relación $T(n) = 2T(\frac{n}{2}) + \mathcal{O}(n)$, y por tanto, $T(n) \in \mathcal{O}(n)$
C. Relación $T(n) = 2T(\frac{n}{2}) + \mathcal{O}(1)$, y por tanto, $T(n) \in \mathcal{O}(n \log n)$
D. Relación $T(n) = 2T(\frac{n}{2}) + \mathcal{O}(n)$, y por tanto, $T(n) \in \mathcal{O}(n \log n)$

Pregunta 2. Con respecto a la complejidad del siguiente algoritmo:

```
def compute(n, a):
    c = 0
    i = 1
    while i < n:
        j = n
        while j > 1:
            c += 1
            j //= 2
        if a % 2 == 0:
            i *= 2
        else:
            i += 1
    return c
```

A. Mejor caso $\mathcal{O}(n\log(n))$; peor caso $\mathcal{O}(\log^2(n))$

B. Mejor caso $\mathcal{O}(n\log(n))$; peor caso $\mathcal{O}(n^2)$

C. Mejor caso $\mathcal{O}(\log^2(n))$; peor caso $\mathcal{O}(n\log(n))$

D. Mejor caso $\mathcal{O}(n)$; peor caso $\mathcal{O}(n\log(n))$

Pregunta 3. ¿Cuál de las siguientes afirmaciones describe el objetivo principal del análisis de algoritmos?

A. Garantizar que un algoritmo sea el más eficiente.

B. Medir el tiempo que un algoritmo tarda en ejecutarse en una máquina.

C. Estudiar el crecimiento de los recursos que consume un algoritmo (como el tiempo de ejecución) en función del tamaño de la entrada.

D. Diseñar algoritmos que sean eficientes en uso de recursos (como el tiempo de ejecución).

Pregunta 4. ¿Por qué el paradigma "divide y vencerás" no es eficiente para resolver problemas como el de la mochila?

A. Porque no divide el problema en subproblemas más pequeños.

B. Porque los subproblemas no son independientes, lo que puede requerir resolverlos varias veces.

C. Porque la combinación de soluciones tiene una complejidad exponencial.

D. Porque no se puede dividir el tamaño de la mochila.

Pregunta 5. ¿Cuál de las siguientes afirmaciones sobre la eficiencia de *Quicksort* y *Mergesort* es correcta?

A. *Quicksort* tiene un mejor caso de $O(n)$ mientras que *Mergesort* siempre tiene $O(n \log n)$.

B. En el peor caso, *Quicksort* puede tener una complejidad de $O(n^2)$, mientras que *Mergesort* garantiza $O(n \log n)$.

C. *Mergesort* es más rápido que *Quicksort* en todos los casos.

D. *Quicksort* siempre tiene una complejidad mejor que $O(n \log n)$.

Pregunta 6. Una empresa compra piezas de oro de n onzas y las trocea en piezas de i onzas ($i = 1, 2, \dots, n$) que luego vende. El corte le sale gratis. El precio de venta de una pieza de i onzas es v_i.

¿Cuál sería la fórmula recursiva que permite calcular la máxima ganancia al vender una pieza de oro de n onzas?

A. $G(n) = \text{máx}_{1 \le i \le n}(v_i + G(n-i)),$ **donde** $G(0) = 0$.

B. $G(n) = \sum_{1 \le i \le n}(v_i + G(n-i)),$ donde $G(0) = 0$.

C. $G(n) = \text{máx}_{1 \le i \le n}(v_i + G(i)),$ donde $G(0) = 0$.

D. $G(n) = \text{mín}_{1 \le i \le n}(v_i + G(n-i)),$ donde $G(0) = 0$.

Pregunta 7. Sea el problema de la mochila discreta (sin posibilidad de fraccionar los objetos), con una capacidad máxima de $W = 10$. Para una instancia concreta, la solución mediante programación dinámica iterativa arroja el siguiente resultado:

Objetos	0	1	2	3	4	5	6	7	8	9	10
0 (sin objetos)	0	0	0	0	0	0	0	0	0	0	0
1 ($v_1 = 8, w_1 = 4$)	0	0	0	0	8	8	8	8	8	8	8
2 ($v_2 = 10, w_2 = 5$)	0	0	0	0	8	10	10	10	10	18	18
3 ($v_3 = 15, w_3 = 8$)	0	0	0	0	8	10	10	10	15	18	18
4 ($v_4 = 4, w_4 = 3$)	0	0	0	4	8	10	10	12	15	18	18
5 ($v_5 = 6, w_5 = 2$)	0	0	6	6	8	10	12	12	15	18	21
6 ($v_6 = 3, w_6 = 1$)	0	3	6	6	9	11	13	13	16	19	21

¿Cuál es una solución completa del problema?

A. [0,0,1,0,1,1]

B. [1,0,0,1,1,1]

C. [0,1,0,1,1,0]

D. [0,0,1,0,1,1]

Pregunta 8. ¿Cuál de las siguientes afirmaciones describe correctamente una diferencia clave entre los enfoques iterativo y recursivo en programación dinámica?

A. El enfoque recursivo siempre es más eficiente en términos de tiempo debido a la reutilización de resultados previamente calculados.

B. El enfoque iterativo evita el coste de la pila de llamadas recursivas, lo que puede ser más eficiente en la práctica.

C. El enfoque iterativo no puede resolver problemas con subproblemas solapados, mientras que el recursivo sí.

D. El enfoque recursivo no requiere una estructura explícita para almacenar resultados intermedios, a diferencia del iterativo.

Pregunta 9. ¿Cuál de las siguientes afirmaciones describe correctamente un algoritmo voraz?

A. Divide el problema en subproblemas independientes y combina las soluciones.

B. Toma decisiones en cada paso que aseguran una solución globalmente óptima.

C. Selecciona en cada paso la mejor opción local, esperando que conduzca a una solución global óptima.

D. Calcula todas las posibles soluciones y elige la mejor.

Pregunta 10. ¿Por qué no se puede resolver el problema de la mochila discreta, sin fraccionar objetos, mediante un algoritmo voraz?

A. Porque no hay una relación directa entre las decisiones óptimas locales y la solución óptima global.

B. Porque requiere dividir el problema en subproblemas independientes.

C. Porque los objetos no pueden ordenarse antes de ir tomando las decisiones.

D. Porque el problema no puede ser representado como un árbol de decisión.

Pregunta 11. ¿Cuál de las siguientes afirmaciones describe correctamente el paradigma de "vuelta atrás"?

A. Es un enfoque para resolver problemas dividiéndolos en subproblemas independientes y combinando sus soluciones.

B. Es una técnica que explora todas las soluciones posibles.

C. Es un método que utiliza una tabla para almacenar resultados intermedios y evitar repetir cálculos.

D. Es una estrategia que selecciona decisiones óptimas en cada paso, sin considerar las futuras.

Pregunta 12. ¿Cuál es el propósito principal de implementar mecanismos de poda en un algoritmo de "vuelta atrás"?

A. Encontrar soluciones más rápidamente ignorando condiciones iniciales.

B. Reducir el número de soluciones posibles que el algoritmo debe explorar.

C. Garantizar que el algoritmo encuentra todas las soluciones posibles.

D. Reducir el tamaño de la entrada para simplificar el problema.

Pregunta 13. ¿Por qué puede ser útil utilizar un algoritmo voraz dentro del esquema de "vuelta atrás"?

A. Para garantizar que la solución encontrada sea la óptima global en todos los casos.

B. Para calcular una cota inicial que permita podar ramas del árbol de búsqueda más rápidamente.

C. Para explorar primero las soluciones más prometedoras.

D. Todas las anteriores son ciertas.

Pregunta 14. En el paradigma de "vuelta atrás", ¿cómo influyen los mecanismos de poda y la inicialización con soluciones voraces en la complejidad del algoritmo?

A. Garantizan que la solución se encuentra de manera más eficiente.

B. Pueden reducir el espacio de búsqueda, lo que lleva a complejidades empíricas menores.

C. Eliminan subárboles de búsqueda, lo que asegura una complejidad lineal.

D. No afectan a la complejidad del algoritmo, ya que el espacio de búsqueda completo debe explorarse en cualquier caso.

Pregunta 15. Una fábrica produce dos tipos de productos: P_1 y P_2. Para fabricar P_1 se necesitan 2 horas de mano de obra y 3 unidades de materia prima, mientras que para P_2 se necesitan 4 horas de mano de obra y 2 unidades de materia prima. La empresa dispone de un máximo de 40 horas de mano de obra y 30 unidades de materia prima por semana. Además, la cantidad producida de P_1 no puede superar el doble de P_2. El beneficio por unidad producida es de 5 euros para P_1 y de 7 euros para P_2. La empresa desea maximizar el beneficio semanal.

Si codificamos el número de P_1 a fabricar como x_1 y el número de P_2 como x_2, ¿cuál de las siguientes es una restricción válida en el modelo de programación lineal para este problema?

A. $2x_1 + 4x_2 \geq 40$

B. $2x_1 + 3x_2 \leq 30$

C. $x_1 - 2x_2 \leq 0$

D. $5x_1 + 7x_2 \leq 40$