

Grado en Ingeniería Informática y Grado en Ingeniería en Tecnologías de la Información

Normas de valoración del examen:

- La nota del examen representa el 80% de la valoración final de la asignatura (el 20% restante corresponde a las prácticas).
 - Cada cuestión contestada correctamente vale 1 punto.
 - Cada cuestión contestada incorrectamente baja la nota en 0.3 puntos.
 - Debe obtenerse un mínimo de 3 puntos en las cuestiones para que el problema sea valorado (con 3 cuestiones correctas y alguna incorrecta el examen está suspenso).
 - La nota total del examen debe ser al menos de 4.5 para aprobar.
 - **Las cuestiones se responden en una hoja de lectura óptica.**
-

Cuestiones:

1. En el hashing cerrado, ¿cuál de las siguientes afirmaciones es cierta?
 - a. Utiliza estructuras dinámicas externas para resolver las colisiones.
 - b. Busca posiciones libres dentro de la tabla.
 - c. Escribe el nuevo valor sobre una posición ocupada.
 - d. Ninguna de las afirmaciones anteriores es cierta.
2. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado con programación dinámica. Supongamos que se dispone de 5 objetos con pesos: 1,3,4,5,7 y beneficios: 2,5,10,14,15 respectivamente, y un volumen máximo de 8. Identifica cuál de las siguientes respuestas correspondería al contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 5, si dichos objetos se consideran en el orden indicado.
 - a. 0 2 2 5 7 7 7 7 7
 - b. 0 2 2 5 10 12 12 15 15
 - c. 0 2 2 5 10 12 14 16 19
 - d. 0 2 2 5 10 14 16 16 19
3. Sea el algoritmo de Dijkstra. Indica cuál de las afirmaciones siguientes es cierta en relación a su coste.
 - a. El coste del algoritmo es independiente de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen.

- b. El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. Pero en el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo es independiente de que el grafo sea disperso o denso.
- c. El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. Pero en el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo si el grafo es disperso está en $O(n^2 \log n)$.
- d. Ninguna de las afirmaciones anteriores es cierta.

4. Elije la afirmación correcta sobre el esquema de Ramificación y Poda:

- a. Los nodos pendientes de visitar se almacenan en una cola con comportamiento “primero en llegar, primero en salir”.
- b. Los nodos pendientes de visitar se almacenan en una pila con comportamiento “primero en llegar, último en salir”.
- c. Los nodos pendientes de visitar se almacenan en un montículo y de esta forma, el recorrido es siempre en anchura.
- d. Ninguna de las otras es cierta.

5. Sobre el esquema de Divide y Vencerás, indicar la respuesta cierta acerca del coste de los algoritmos derivados de dicho esquema:

- a. El coste que se consigue es siempre igual o más eficiente que $O(\log n)$.
- b. El coste que se consigue es siempre igual o más eficiente que $O(n \log n)$.
- c. El coste no puede ser en ningún caso exponencial, es decir, de tipo $O(2^n)$.
- d. Ninguna de las otras es cierta.

6.Cuál de las siguientes cuestiones es cierta:

- a. La creación de un montículo a partir de un vector de valores, puede tener coste lineal.
- b. Los montículos de máximos son equivalentes a un árbol binario de búsqueda, con los elementos menores que la raíz en el subárbol izquierdo y los mayores en el derecho.
- c. Un algoritmo voraz siempre encuentra la solución óptima en tiempo lineal, es decir en un tiempo proporcional al tamaño del conjunto de candidatos.
- d. Ninguna de las otras respuestas es correcta.

PROBLEMA

Partiendo de un conjunto $N=\{n_1, n_2, \dots, n_m\}$ compuesto por m número positivos y de un conjunto $O=\{+, -, *, /\}$ con las operaciones aritméticas básicas, se pide obtener una secuencia de operaciones factible para conseguir un número objetivo P . Como restricciones al problema, debe tenerse en cuenta que:

- a. Los números del conjunto N pueden utilizarse en la secuencia de operaciones 0 o 1 vez,
- b. los resultados parciales de las operaciones pueden utilizarse como candidatos en operaciones siguientes,
- c. las operaciones que den como resultado valores negativos o números no enteros NO deberán tenerse en cuenta como secuencia válida para obtener una solución.

Diseñe un algoritmo que obtenga una solución al problema propuesto, mostrando la secuencia de operaciones para obtener el número objetivo P . En caso de no existir solución alguna, el algoritmo deberá mostrar la secuencia de operaciones que dé como resultado el valor más próximo, por debajo, del número objetivo P . Por ejemplo, siendo $P=960$ y $N=\{1,2,3,4,5,6\}$, la secuencia de operaciones que obtiene la solución exacta es: $(((((6*5)*4)*2)*(3+1)))=960$. Si $P=970$, el algoritmo no encontraría la solución exacta con el conjunto de números inicial y la secuencia más próxima por debajo de P sería $(((((6*5)*4)*2)*(3+1)))=960$.

La resolución del problema debe incluir, por este orden:

1. Elección del esquema más apropiado, el esquema general y explicación de su aplicación al problema (0,5 puntos).
2. Descripción de las estructuras de datos necesarias (0,5 puntos solo si el punto 1 es correcto).
3. Algoritmo completo a partir del refinamiento del esquema general (2,5 puntos solo si el punto 1 es correcto). Si se trata del esquema voraz, debe realizarse la demostración de optimalidad. Si se trata del esquema de programación dinámica, deben proporcionarse las ecuaciones de recurrencia.
4. Estudio del coste del algoritmo desarrollado (0,5 puntos solo si el punto 1 es correcto).