



Guía 2

Árboles, hashing, orden lineal y backtracking

1. Árboles de búsqueda

1. (12.2-4[CLRS09]) Suponga que la búsqueda de la llave k en un ABB termina de forma exitosa en una hoja. Llamaremos *camino de búsqueda* a la colección de llaves que fueron comparadas hasta llegar a k , incluyéndola. Considere tres conjuntos:

- A : llaves a la derecha del camino de búsqueda
- B : llaves del camino de búsqueda
- C : llaves a la izquierda del camino de búsqueda

Entregue el contraejemplo más pequeño que pueda para la siguiente afirmación:

$$\forall a \in A, \forall b \in B, \forall c \in C. \quad a \leq b \leq c$$

2. (I2-2018-2)

- a) Muestre que cualquier árbol binario de búsqueda (no necesariamente balanceado) puede ser transformado en cualquier otro árbol binario de búsqueda con las mismas claves mediante una secuencia de rotaciones simples.
- b) Determine un orden en que hay que insertar las claves 1, 3, 5, 8, 13, 18, 19 y 24 en un árbol 2-3 inicialmente vacío para que el resultado sea un árbol de altura 1, es decir, una raíz y sus hijos.
- c) Considere un árbol rojo-negro formado mediante la inserción de n nodos usando el procedimiento visto en clase. Justifique que si $n > 1$, el árbol tiene al menos un nodo rojo.
- d) Muestre cómo construir un árbol rojo-negro que demuestre que, en el peor caso, casi todas las rutas desde la raíz a una hoja tienen largo $2 \log(n)$, en que n es el número de nodos del árbol.

3. (I2-2022-1)

- a) Considere una secuencia de inserciones de claves (distintas) que se ejecuta tanto en un árbol AVL como en un árbol rojo-negro, ambos inicialmente vacíos. La secuencia es tal que mantiene los árboles balanceados tanto como sea posible. ¿Cuál de los dos árboles se desbalancea primero (de acuerdo con las reglas/propiedades de balance del árbol correspondiente)? Explique, justifique o demuestre claramente su respuesta.
- b) Dibuje un árbol rojo-negro (es decir, que cumpla todas las propiedades de árbol rojo-negro), pero tal que, si nos olvidamos de los colores de los nodos, no sea un árbol AVL (es decir, que no cumpla alguna de las propiedades de árbol AVL). Explique, justifique o demuestre claramente su respuesta.
- c) Demuestre o justifique claramente que dado cualquier árbol AVL (es decir, que cumpla todas las propiedades de árbol AVL), sus nodos siempre pueden ser pintados de manera tal que el árbol sea un árbol rojo-negro (es decir, que cumpla todas las propiedades de árbol rojo-negro).

2. Tablas de hash

1. (11.2-2[CLRS09]) Considere una tabla de hash T de tamaño $m = 9$ que maneja colisiones mediante encadenamiento, y que opera con la función de hash $h(k) = k \bmod 9$. Muestre qué sucede cuando se insertan las llaves 5, 28, 19, 15, 20, 33, 12, 17 y 10, en ese orden.
2. (11.2-3[CLRS09]) Suponga que modificamos el esquema de encadenamiento para manejo de colisiones, de manera que en cada inserción las listas ligadas se mantienen ordenadas (con su algoritmo de ordenación favorito). ¿Cómo cambia el tiempo de ejecución de las búsquedas exitosas, no exitosas, inserciones y eliminaciones?
3. (I2-2022-1) La Universidad ha desarrollado una aplicación que permite verificar en línea si un estudiante (utilizando como llave su número de alumno de 7 cifras sin dígito verificador) es o no un alumno regular de la Universidad. Para dicha verificación utiliza un ABB donde se mantienen los datos de los aproximadamente 25.000 alumnos regulares y se realizan las búsquedas, inserciones y eliminaciones necesarias para mantener este conjunto actualizado (del orden de 7.000 actualizaciones por semestre).

Esta aplicación funciona de manera adecuada para volúmenes acotados de consultas simultáneas, por ejemplo, en la verificación de préstamos en bibliotecas. Sin embargo, se desea ampliar su aplicación al control de ingreso a los campus, donde el nivel de masividad y concentración en un corto período de tiempo obliga a optimizar al máximo la velocidad de búsqueda de la aplicación.

- a) Proponga una mejora en la estructura de datos utilizada en el sistema considerando la necesidad de optimizar el desempeño de las búsquedas y las necesidades de actualización indicadas. Justifique por qué su solución es mejor que el ABB.
- b) Escriba en pseudocódigo los algoritmos de búsqueda, inserción y eliminación para su propuesta.
- c) La implementación de su mejora requiere mover toda la información del sistema original (ABB) al sistema diseñado por Ud. Proponga en pseudocódigo un algoritmo que permita llevar todos los datos del ABB original a su solución. Puede utilizar los algoritmos de su solución para (b) en su respuesta.

3. Ordenación lineal

1. (8.2-4[CLRS09]) Describa un algoritmo que, dados n enteros en el rango 0 hasta k , preprocese el input y luego permita responder en tiempo $\mathcal{O}(1)$ la consulta de cuántos de los n números están en el rango $[a \dots b]$. Su algoritmo debe usar tiempo de preprocesamiento $\Theta(n + k)$.
2. (I2-2022-1) Una central telefónica registra en un archivo LOG las llamadas realizadas a través de ella durante un mes dado. En dicho archivo, en cada fila, registra:

`hora_llamada, fecha_llamada, hora_llamada, número_origen, número_destino, duración_llamada.`

Se desea utilizar el archivo LOG para realizar la facturación del servicio telefónico, indicando separadamente para cada número de origen el detalle en orden cronológico de cada llamada realizada (fecha, hora, destino, duración) y el costo total del servicio calculado como la suma de las duraciones de las llamadas multiplicado por el valor del minuto.

- a) Proponga, utilizando la materia vista en el curso hasta el momento, una solución para obtener la facturación indicada, de la manera más eficiente en tiempo posible.
- b) Muestre que su solución cumple con los requisitos indicados para cada característica de la facturación.
- c) Un compañero le propone usar quickSort en LOG; luego separar por origen para construir cada factura. Compare su solución con la propuesta por su compañero en términos de eficiencia.

4. Backtracking

1. **(I2-2022-1)** La siguiente matriz representa un pasillo por el cual debe avanzar un pequeño robot desde el lado izquierdo al derecho. El tamaño del robot es de una celda (1x1). El pasillo tiene celdas inseguras para el robot (marcadas por un 0), las que debe evitar junto a las adyacentes a estas (izquierda, derecha, arriba y abajo) para avanzar con seguridad. Los movimientos permitidos para el robot son sólo por fila o columna (no se permite avanzar en diagonal).

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

En este escenario, se requiere determinar para cualquier matriz de tamaño dado y un número arbitrario de celdas inseguras distribuidas en la matriz, una ruta segura para ir de izquierda a derecha, partiendo desde cualquier celda de la primera columna de la izquierda y llegando a la última columna de la derecha (en cualquiera de sus celdas).

- a) Identifique variables, dominios y restricciones del problema planteado.
 - b) Proponga un algoritmo en pseudocódigo para resolver el problema (marcar el camino con una X), indicando el largo de la ruta encontrada (número de celdas que recorre el camino).
 - c) Modifique su algoritmo de (b) para encontrar la ruta más corta.
2. **(I1-2017-2)** En el problema de programación de máquinas, tenemos n tareas que realizar, cada una con su hora de inicio s_i y su hora de finalización f_i , y tenemos un número suficientemente grande de máquinas para realizar las tareas. Queremos realizar todas las tareas usando el **menor número posible** de máquinas, respetando la restricción de que una máquina solo puede realizar una tarea a la vez. Es decir, dos tareas se pueden realizar en una misma máquina solo si la hora de finalización de una es menor o igual que la hora de inicio de la otra. Dado un conjunto de tareas $T = \{t_1, \dots, t_n\}$ y un conjunto de máquinas $M = \{m_1, \dots, m_n\}$, proponga un algoritmo para resolver este problema.

5. Referencias

Referencias

- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.