CC40A - Guía 5

Arboles de búsqueda

1. Diseñe un algoritmo de tiempo $O(\log n)$ promedio para unir dos árboles binarios A_1 y A_2, si se sabe que todas las claves de A_1 son menores que las de A_2
2. Haga lo mismo para el caso de que los árboles sean 2-3.
3. Diseñe un algoritmo que dado un árbol de altura h pueda contestar en tiempo $O(h)$ cuál es el número que sigue a x almacenado en el árbol. Es decir, si en el árbol hay 1, 5 y 8, el siguiente de 3 es el 5.
4. Calcule el tamaño óptimo de la hoja de un árbol B para disco. Tenga en cuenta que para accesar una hoja en disco hay un tiempo de seek promedio t_s, una latencia promedio t_ℓ y un tiempo de transferencia que depende de la cantidad de información a transferir (expresado como t_t por byte transferido a memoria). Optimice el tiempo de búsqueda suponiendo que las hojas están 70% llenas (es decir se accesan $\log_{m*0.7} n$ hojas).
5. Diseñe una estructura de datos que además de las operaciones usuales permita responder, también en $O(\log n)$ peor caso, la pregunta: "cuántos elementos $\geq x$ hay en el conjunto?".
6. Diseñe una estructura de datos que almacene elementos como siempre, junto con un puntaje (cero al insertar). Además de poder recuperar el puntaje de un elemento cuando se lo busca, se desea poder sumar un valor a a *todos* los elementos del conjunto que estén entre los valores x e y inclusive. Todo en tiempo $O(\log n)$ peor caso.

Hashing

1. El hashing con *listas mezcladas* es una mezcla entre cerrado y abierto. Todo se guarda en la tabla, pero ahí se almacena, además de los elementos, un puntero (posición de la tabla) al siguiente de la lista. Es decir, se usa la misma tabla como almacenamiento para las listas de colisiones. Note que al "alocar" una celda debido a una colisión se puede estar ocupando la celda que podría querer ocupar otro elemento por su función de hashing. Explique cómo funciona este esquema, describa las operaciones de búsqueda, inserción y borrado, y compárelo con los otros. Vale la pena?
2. Compare la eficiencia de hashing abierto y cerrado considerando que almacena enteros (que miden lo mismo que un puntero).
3. Describa un mecanismo compacto para almacenar los nodos internos del hashing extendible. Cuánto espacio requiere para n hojas? Si cada hoja es una página de disco de 4K y almacena enteros de 4 bytes, y las hojas están llenas al 60%, cuántos enteros puede almacenar manteniendo los nodos internos de la estructura en una memoria principal de 16 Mb ?

4. Una forma de agilizar la búsqueda en hashing extendible es no ir mirando la función de hashing bit a bit, sino que cuando el "trie" de bits está completo en, digamos, los siguientes 3 niveles, hacer un array de 8 posiciones que se indexan con los siguientes 3 bits de la función de hashing. Explique cómo modificar la búsqueda, inserción y borrado y comente las ventajas y desventajas de esta estructura.

5. Compare ventajas y desventajas de árboles B versus hashing extendible. Entre las operaciones de interés incluya imprimir el conjunto entero en forma ordenada.

Tries

1. Para reducir el espacio del trie se considera ver a los caracteres como secuencias de 8 bits, y hacer el trie suponiendo que tenemos un alfabeto binario. Comente las ventajas y desventajas de este esquema.

2. Explique cómo buscar la mínima o máxima cadena almacenada en un trie (bajo el orden lexicográfico). Entre las dos variantes de almacenamiento de hijos que vimos, cuál resulta más eficiente?

3. Se considera la alternativa de almacenar strings en los nodos internos del trie en vez de hacerlo sólo en las hojas. Explique cómo podría funcionar ésto y describa los algoritmos de búsqueda, inserción y borrado. Vale la pena hacer un árbol Patricia bajo este esquema?

4. Escriba detalladamente los algoritmos para buscar, insertar y borrar en tries, en los árboles propuestos por ustedes en la clase y para árboles patricia.

Union-find

Muestre que cuando se usa el truco de colgar el árbol más chico del más grande, también se puede usar la altura del árbol como indicador de tamaño en vez del número de nodos, y el lema sigue valiendo. Cuál puede ser la complicación de esto al usar compresión de caminos?