AED Tema 3. Árboles II: estructuras para búsqueda

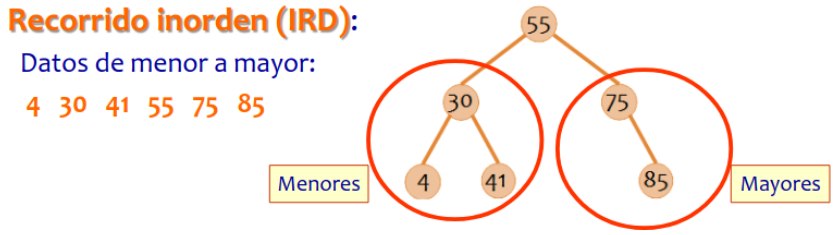
Martín González Dios

2 de enero de 2025

1. Árboles binarios de búsqueda (ABB)

1.1. Definición

- **Dado un nodo**: todos los datos del subárbol izquiero son menores, todos los datos del subárbol derecho son mayores.
- **Clave de ordenación**: criterio de ordenación menor - mayor.
- Ventaja: **tiempo de ejecución promedio O(log(N))**. Pero es necesario imponer condiciones de balanceado (árbole AVL, árboles B, etc) sobre los ABB para que se mantenga O(log(N)), ya que si no balanceamos en el peor caso O(N) de tiempo de ejecución.
- Recorrido inorden (IRD): datos de menor a mayor.



1.2. Inserción

En un ABB la inserción no se puede realizar en el primer hueco libre que se vea, sino que hay que **respetar la estructura**, buscando desde la raíz el hueco correspondiente al nodo que queremos insertar. Por tanto, la inserción es recursiva, ya que se parte del nodo raíz y se va mirando si el elemento que queremos insertar es menor o mayor, bajando por la rama correspondiente hasta encontrar su sitio.

Como clara **desventaja** de este tipo de árbol tenemos el caso en el que la mayor parte de los elementos que se inserten sean mayores mayores o menores que el nodo raíz, provocando un **árbol desbalanceado**, lo que aumenta el tiempo de búsqueda al tener que recorrer muchos elementos.

1.3. Eliminación

La desventaja en el caso de la eliminación reside en tener que **reestructurar el árbol si se eliminan ciertos nodos**, para mantener la estructura que cumpla las restricciones de los ABB.

1. Si el **nodo es hoja**, se suprime del árbol sin más.
2. Si el nodo tiene **un único hijo**, se intercambia de posición con su hijo y se elimina del árbol.
3. Si el **nodo tiene dos hijos**, buscamos el menor de todos los descendientes del hijo derecho, sustituimos por dicho nodo, y lo eliminamos del árbol. Otra opción es sustituir por el nodo de mayor valor de los descendientes del subárbol izquierdo.

2. Árboles parcialmente ordenados o montículos

Es un **árbol binario completo parcialmente ordenado**:
Completo: en el nivel más bajo pueden faltar algunos nodos, pero éstos han de ser los más a la derecha posibles.
Parcialmente ordenado, existen 2 tipos:

- **Montículo de mínimos**: clave nodo $\leq$ clave hijos (padre $<$ que hijos).
- **Montículo de máximos**: clave nodo $\geq$ clave hijos (padre $>$ que hijos).

2.1. Propiedades del montículo binario o heap

- Da soporte eficiente a los **operadores del TAD cola de prioridad**, que proporciona una flexibilidad extra a la hora de ordenar.
- El orden parcial es más débil que el orden total (sin embargo es más eficiente de mantener), pero más fuerte que el orden aleatorio (de manera que el elemento con mayor prioridad se puede identificar de forma rápida, ya que siempre es la raíz).

2.2. Inserción

Como árbol binario completo que es, la inserción se hace en el **hueco libre más a la izquierda de su último nivel**. Sin embargo, si se inserta un **elemento** con un valor (para este ejemplo considérese montículo de mínimos) **menor que su padre**, es necesario **reorganizar el árbol** intercambiando el elemento con su padre hasta llegar a una posición en la que se cumpla el criterio.

2.3. Eliminación

En este tipo de árboles, el elemento que se quiere **suprimir es siempre el de mayor prioridad**, es decir, **la raíz**. Para llevar a cabo esta operación, el nodo raíz se intercambia de posición con el nodo hoja más a la derecha (el nodo más a la derecha del último nivel), y una vez allí, se elimina. A continuación, se procede a la reorganización del nuevo nodo raíz hasta encontrar su sitio, intercambiándolo con (en nuestro caso) el hijo menor en cada caso.

2.4. Monticulización (Heapify)

Proceso en el que dado un conjunto de elementos, nos pedirán **representarlos por medio de un montículo** (de máximos o de mínimos).

1. **Dibujar el árbol** que resulta a partir del array.
2. Si el árbol tiene L niveles, **desde i = L - 1 hasta 1**:
   - Comprobar si cada nodo de nivel i cumple la **regla de orden parcial**. Si no la cumple intercambiar con el menor/mayor de sus hijos.
   - **Propagar esta comprobación** hacia los niveles desde i hasta L - 1.