



Tema 5 - Árboles

Guía Docente

Las estructuras lineales son muy útiles a la hora de representar los datos en nuestro programa, sin embargo no sirven para representar datos que están relacionados de una forma jerárquica (por ejemplo, estructura de documentos, organigramas o árboles genealógicos). Este tema está dedicado al estudio de un nuevo tipo de datos, los árboles, que nos permitirá representar información donde los datos se relacionan con estructuras jerárquicas.

El tema comienza presentando los conceptos generales de un árbol: raíz, grado de un árbol, nodos internos y externos, profundidad, altura, etc. También se describen los algoritmos para recorrer (visitar) los elementos de un árbol. El tema continúa con la implementación de un árbol binario y sus operaciones básicas, es decir, sus recorridos y operaciones básicas como calcular la altura o el tamaño.

A continuación, se presenta el tipo abstracto de datos de árbol binario de búsqueda. Este nuevo tipo permite representar secuencias de elementos de una forma más eficiente que las estructuras lineales, ya que la complejidad de las operaciones de búsqueda, inserción y borrado son logarítmicas. Cada una de las operaciones son presentadas primero de forma intuitiva por medio de ejemplos sobre árboles concretos y después su implementación es explicada paso a paso. Además, se explica por qué la complejidad temporal de sus operaciones búsqueda, inserción y borrado es logarítmica, siendo más eficiente que la complejidad proporcionada por las listas (lineal).

También se explica por qué los árboles deben estar balanceados en altura (las ramas izquierda y derecha de cada nodo tienen que tener una altura similar. Si el árbol está desbalanceado la complejidad de sus operaciones básicas se convierte en lineal, perdiendo así la principal ventaja de los árboles binarios de búsqueda frente a las listas.

Se estudian los árboles AVL, un nuevo tipo de árbol binario de búsqueda propuesto por dos matemáticos Adelson-Velskii y Landis en 1962. Estos árboles se caracterizan por implementar rotaciones en sus operaciones de inserción y borrado para conseguir que los árboles siempre estén equilibrados en altura, asegurando así la complejidad logarítmica en sus operaciones básicas. Además de estudiar cada

una de las rotaciones con ejemplos, también se presenta su implementación en Python.

El material se completa con dos hojas de problemas que permitirá practicar a los estudiantes los principales conceptos estudiados durante este tema.

Al final del tema, los estudiantes deberían ser capaces de:

1. Comprender el concepto de árbol y sus principales propiedades.
2. Ser capaces de implementar una estructura de datos que permita representar un árbol binario e implementar métodos para calcular sus propiedades así como para recorrer los nodos del árbol, en concreto, se estudiarán los recorridos in-order, pre-order, post-order y por niveles.
3. Ser capaces de implementar una estructura de datos que permita representar un árbol binario de búsqueda e implementar sus operaciones básicas de búsqueda, borrado e inserción.
4. Comprender por qué la complejidad de un árbol binario de búsqueda es logarítmica, y qué deben cumplir los árboles binarios de búsqueda para que complejidad sea siempre logarítmica.
5. Conocer el concepto de nodo balanceado y calcular su factor de equilibrio.
6. Conocer el concepto de árbol AVL o equilibrado.
7. Conocer y saber aplicar las principales rotaciones de los árboles AVL.
8. Ser capaces de implementar las operaciones de inserción y borrado en un árbol AVL.
9. Ser capaces de implementar nuevas operaciones en las estructuras de datos estudiadas.
10. Ser capaces de extender y combinar las diferentes estructuras en el curso para resolver problemas más complejos.
11. Resolver problemas utilizando árboles.