

ESTRUCTURAS DE DATOS

TRABAJO PRÁCTICO N ° 10

Árboles de búsqueda: AVL, Trie y Árboles 2/3

*Departamento de Ciencias e Ingeniería de la Computación - U.N.S.
Primer cuatrimestre de 2019*

Bibliografía:

[GT] Michael Goodrich & Roberto Tamassia. *Data Structures and Algorithms in Java. Fourth Edition*. John Wiley and Sons, 2006.

Sergio A. Gómez. *Estructuras de datos, notas del curso*, 2019.

Ejercicio 1: Árboles AVL (Adel'son -Vel'skii y Landis)

- Defina árbol AVL.
- Indique cuál es la diferencia importante de estos árboles con respecto a los árboles binarios de búsqueda.
- ¿Para qué se utilizan rotaciones en estos árboles? Grafique las cuatro rotaciones.
- Escriba en Java una clase que implemente un mapeo usando un árbol AVL. Proponga implementaciones para los métodos get y put.
- Implemente remove usando una eliminación "lazy" (i.e. con una marca de borrado en los nodos).
- Escriba en Java un método que reciba un árbol binario, e indique si es AVL o no. Note que este árbol no tiene el campo altura como atributo de los nodos.
- Escriba las operaciones insertar de manera que se mantenga la propiedad de los árboles AVL. Analice el orden del tiempo de ejecución.
- Implemente el TDA mapeo utilizando un árbol AVL. Calcule el orden del tiempo de ejecución de las operaciones.

Ejercicio 2: Árboles (2,3)

- Defina la noción de árbol (2,3).
- Indique cuál es la diferencia importante de estos árboles con respecto a los árboles binarios de búsqueda.
- Analice el tiempo de ejecución de las operaciones de un TDA diccionario implementado con árboles 2-3.
- Analice el tiempo de ejecución de las operaciones de un TDA mapeo implementado con árboles 2-3.
- Proponga una definición de clase para el mapeo utilizando un árbol (2,3), implemente solamente el constructor y la operación Recuperar.
- Sea D un diccionario cuyos elementos tienen llave de tipo entero.Cuál sería la estructura de datos en Java para implementar un diccionario utilizando la estructura de árbol 2-3.
- Dibuje un árbol 2-4 para representar el conjunto { 5, 10, 12, 15, 18, 20, 21, 23, 30 }. Luego dibuje el árbol resultante de insertar el elemento 22. Por último dibuje el árbol que resulta de eliminar los elementos 5 y 10 (en ese orden).

Ejercicio 3: TRIEs (re-trie-val trees)

- a) Defina un árbol TRIE. ¿Cuál es el objetivo de este tipo de estructura de datos?
- b) Especifique claramente cuál es el tipo elemento de un árbol trie.
- c) Implemente un Trie con arreglos para los nodos.
- d) Considere una variante de Trie donde la lista de hijos se representa con una lista en lugar de un arreglo como se vio en la teoría. Compare ambas implementaciones en términos de los requerimientos a de memoria y eficiencia de las operaciones.

Ejercicio 4:

- a) Implemente un mapeo de palabras en un tipo genérico V utilizando un trie. Calcule el orden del tiempo de ejecución de las operaciones Get y Put.
- b) Sea M un mapeo implementado con un árbol TRIE. Escriba en Java un procedimiento que devuelva la palabra más larga almacenada en el mapeo M. Indique el orden del tiempo de ejecución del procedimiento implementado justificando adecuadamente.

Ejercicio 5: (trie)

Sea D un conjunto de strings implementado con un árbol TRIE. Escriba en Java un método que cuente la cantidad de palabras del dominio del conjunto que terminan con "ABLE" (como por ejemplo adorable, usable, hable,...). Indique el orden del tiempo de ejecución del procedimiento implementado. Justifique