



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

Guía de Ejercicios Arboles

Algoritmos y Estructura de datos

Curso Ing. Gustavo Schmidt

Árboles AVL

Nivel Básico

1. **Inserción básica:**
 - Inserta los números: 10, 20, 30, 40, 50, 11, 21, 31, 41, 51, 12, 22, 32, 42, 52, 13, 23, 33, 43, 53 en un árbol AVL. Realiza y documenta los reequilibrios necesarios.
2. **Eliminación simple:**
 - Inserta los números: 15, 10, 20, 25, 115, 110, 120, 125. Luego, elimina el número 10. Explica cómo se rebalancea el árbol.
3. **Contar hojas:**
 - Inserta los números: 5, 3, 8, 7, 10. Escribe un algoritmo para contar cuántas hojas tiene el árbol resultante.

Nivel Intermedio

4. **Inserción con múltiples rotaciones:**
 - Inserta los números: 50, 25, 75, 10, 30, 60, 80. Documenta cada rotación necesaria.
5. **Eliminación con casos complejos:**
 - Crea un árbol AVL insertando los números: 40, 20, 60, 10, 30, 50, 70. Luego elimina el número 20 y analiza el proceso.
6. **Altura del árbol:**
 - Implementa un algoritmo para calcular la altura de un árbol AVL después de insertar los números: 18, 15, 30, 25, 40.

Nivel Avanzado

7. **Secuencia larga de operaciones:**
 - Inserta los números: 100, 50, 150, 25, 75, 125, 175. Luego elimina los números 50 y 150. Explica cómo cambia la estructura del árbol en cada paso.
 8. **Validación de AVL:**
 - Diseña un algoritmo para verificar si un árbol dado es un árbol AVL válido.
 9. **Camino más largo:**
 - Implementa un algoritmo para encontrar el camino más largo desde la raíz hasta una hoja en un árbol AVL.
-

Árboles B

Nivel Básico

1. **Construcción básica:**
 - Construye un árbol B de grado 3 insertando los números: 10, 20, 30, 40, 50. Muestra los pasos de división de nodos.
2. **Eliminación simple:**

- Inserta los números: 15, 10, 20, 25. Luego elimina el número 15. Explica cómo cambia la estructura.

3. Recorrido en orden:

- Inserta los números: 8, 4, 12, 10, 14. Implementa un algoritmo para realizar un recorrido en orden del árbol resultante.

Nivel Intermedio

4. Inserción con divisiones múltiples:

- Inserta los números: 1, 2, 3, 4, 5, 6, 7, 8 en un árbol B de grado 3. Muestra todos los pasos intermedios.

5. Eliminación compleja:

- Inserta los números: 50, 25, 75, 10, 30, 60, 80. Elimina el número 75. Analiza y explica los cambios en los nodos.

6. Altura del árbol:

- Diseña un algoritmo para calcular la altura de un árbol B después de insertar: 5, 15, 25, 35, 45, 55.

Nivel Avanzado

7. Secuencia larga de operaciones:

- Construye un árbol B de grado 4 insertando los números: 100, 200, 300, 400, 500, 600, 700, 800, 900. Luego elimina 300 y 500. Explica las transformaciones.

8. Validación de árbol B:

- Implementa un algoritmo para verificar si un árbol es un árbol B válido de un grado dado.

9. Nodos internos y hojas:

- Escribe un algoritmo para contar la cantidad de nodos internos y hojas en un árbol B.

Árboles B+

Nivel Básico

1. Construcción básica:

- Crea un árbol B+ de grado 3 insertando los números: 10, 20, 30, 40, 50. Muestra los pasos para estructurar los nodos hoja.

2. Recorrido en orden:

- Construye un árbol B+ de grado 3 con los números: 5, 15, 25, 35, 45, 55. Implementa un algoritmo para recorrer todos los elementos en orden ascendente.

3. Contar nodos hoja:

- Después de insertar: 10, 20, 30, 40, 50, escribe un algoritmo para contar los nodos hoja.

Nivel Intermedio

4. Inserción con divisiones:

- Inserta los números: 1, 2, 3, 4, 5, 6, 7, 8 en un árbol B+ de grado 4. Muestra cómo se reorganizan los nodos hoja y los índices.
- 5. **Eliminación básica:**
 - Construye un árbol B+ de grado 3 insertando: 15, 30, 45, 60, 75, 90. Luego elimina 30 y analiza cómo cambia el árbol.
- 6. **Cargar desde una lista:**
 - Implementa un algoritmo para construir un árbol B+ a partir de una lista de números: [18, 12, 30, 45, 7, 22, 50].

Nivel Avanzado

- 7. **Eliminación con reorganización compleja:**
 - Construye un árbol B+ de grado 4 insertando: 100, 200, 300, 400, 500, 600, 700. Luego elimina 300 y 400. Explica los cambios.
- 8. **Diseño eficiente de consulta:**
 - Diseña un algoritmo para buscar un rango de valores, por ejemplo, [30, 90], en un árbol B+.
- 9. **Validación de árbol B+:**
 - Implementa un algoritmo para verificar si un árbol es un B+ válido, comprobando las relaciones entre nodos internos y hojas.