Enma Leticia Ramírez Castro

Carnet: 0907-23-18206

1) **Comparación entre la versión BST y la versión AVL**

   Cuando se comparan los arboles BST Y AVL esperamos a que el Árbol AVL tenga tiempos de operación en rendimiento en volúmenes grandes de datos. Esto sucede porque el árbol AVL se auto-balancea después de cada inserción o eliminación, asegurando que la altura del árbol siempre se mantenga dentro de un límite razonable de logaritmos respecto a los nodos.

   Con respecto a BST no tiene ninguna restricción de balanceo, lo que lo convierte en un árbol desbalanceado si los datos son insertados en orden que favorezca el desbalance, como por ejemplo que insertemos elementos o valores en orden ascendente o descendente.

2) **Porque el árbol AVL tiende a ser más rápido en operaciones de búsqueda, especialmente con grandes volúmenes de datos**

   El árbol AVL mantiene su altura con equilibrio lo que permite que este en un rango O()log n. Esto da como significado que la cantidad de comparaciones necesarias para encontrar un valor es relativamente baja, incluso con grandes volúmenes de datos.

   El balanceo constante evita que el árbol crezca de manera sesgada.

   En cambio, BST se vuelve ineficiente cuando se encuentra desbalanceado y la búsqueda puede requerir recorrer todos los nodos de una cadena, como cuando un árbol se convierte en una lista enlazada.

3) **Casos con más diferencia**

   Entre BST y AVL se muestra en los casos de inserciones y búsquedas cuando los datos son ingresados ordenadamente o en secuencia. Si los elementos los insertamos de manera ascendente o descendente, un BST se desbalanceará rápidamente, degenerando en una lista lo que incrementará en el tiempo de búsqueda a O(N). En cambio, en un árbol AVL mantendrá siempre un balance adecuado y el tiempo de búsqueda seguirá siendo O (log n)

4) ¿Hubo operaciones donde BST fue más rápido? ¿Por qué podría haber ocurrido?

   - ✓ Si el árbol BST esta balanceado desde el principio o tiene pocos nodos, el costo adicional de mantener el balance en un árbol AVL no es necesario. Las operaciones de búsqueda, inserción y eliminación en BST podrían ser más rápidas porque no hay necesidad de recalcular y ajustar las alturas de los nodos ni realizar las rotaciones necesarias para la mantención del balance.
   - ✓ En un bajo número de rotaciones si BST tiene una distribución de datos que ya está en un estado equilibrado la operación de inserción puede ser más rápida que en un árbol AVL porque no hay necesidad de hacer tantas rotaciones para balancear el árbol después de cada inserción.

Resumen: El árbol AVL suele ser más rápido especialmente con grandes cantidades de datos, ya que tiene capacidad de mantener el equilibrio automático, que le asegura el rendimiento de O (log n) para las operaciones de búsqueda, inserción y eliminación.

Estructura del programa

Proyecto llamado ConversionBSLpor AVL

Almacenamiento de árbol AVL

Arbol.cs

ArbolAVL.cs

NodoAVL.cs

Medicion de tiempo

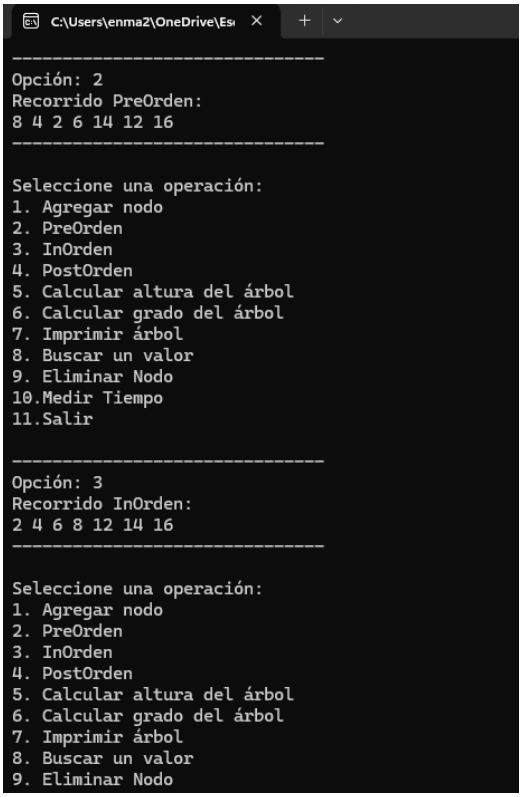
arbolBST.cs

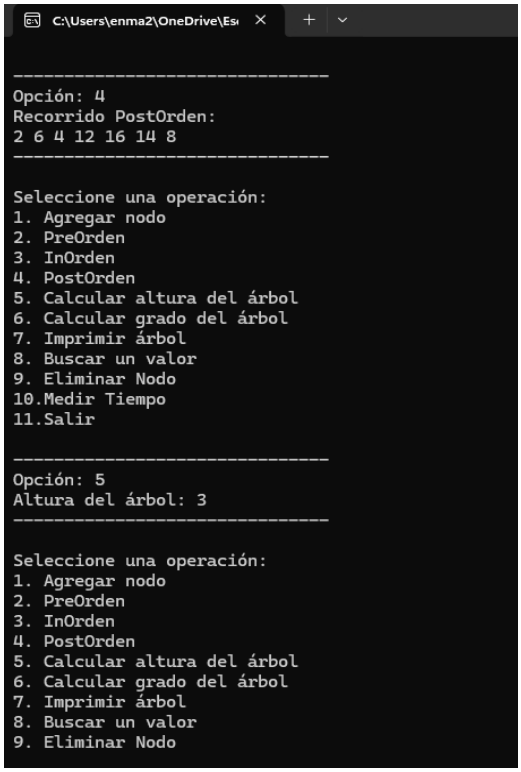
arbolAVL.cs

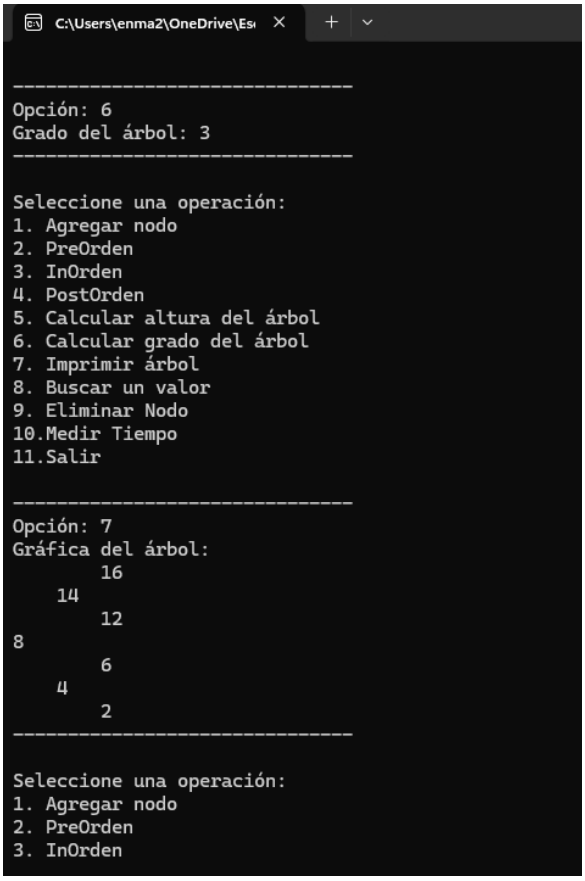
Nodo.cs

Program.cs

Captura de código en su estructura ArbolVAL


C:\Users\enma2\OneDrive\Es
Opción: 2
Recorrido PreOrden:
8 4 2 6 14 12 16
Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden
4. PostOrden
5. Calcular altura del árbol
6. Calcular grado del árbol
7. Imprimir árbol
8. Buscar un valor
9. Eliminar Nodo
10.Medir Tiempo
11.Salir
Opción: 3
Recorrido InOrden:
2 4 6 8 12 14 16
Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden
4. PostOrden
5. Calcular altura del árbol
6. Calcular grado del árbol
7. Imprimir árbol
8. Buscar un valor
9. Eliminar Nodo


C:\Users\enma2\OneDrive\Es
Opción: 4
Recorrido PostOrden:
2 6 4 12 16 14 8
Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden
4. PostOrden
5. Calcular altura del árbol
6. Calcular grado del árbol
7. Imprimir árbol
8. Buscar un valor
9. Eliminar Nodo
10.Medir Tiempo
11.Salir
Opción: 5
Altura del árbol: 3
Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden
4. PostOrden
5. Calcular altura del árbol
6. Calcular grado del árbol
7. Imprimir árbol
8. Buscar un valor
9. Eliminar Nodo

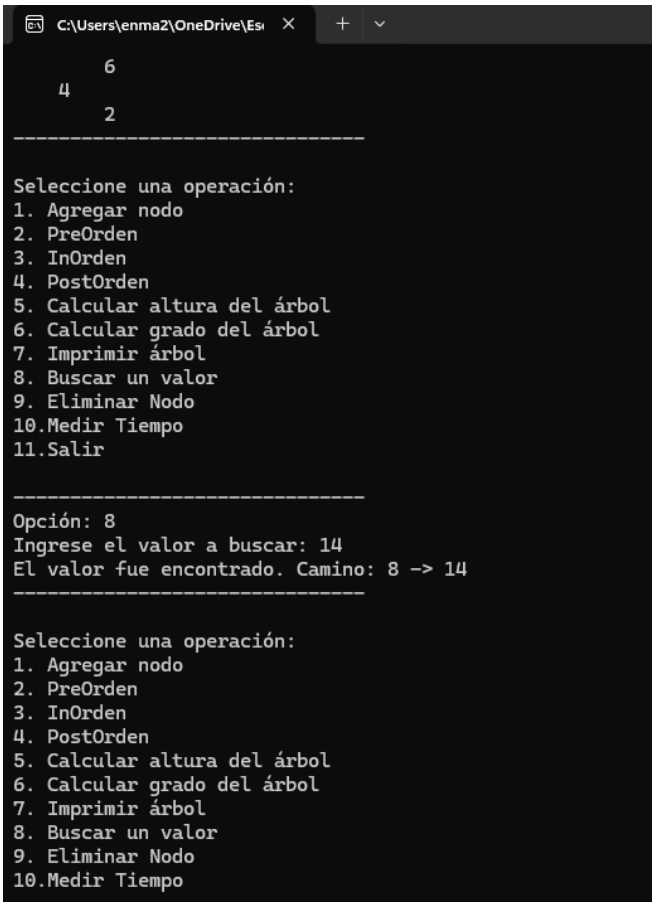

C:\Users\enma2\OneDrive\Es

Opción: 6
Grado del árbol: 3

Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden
4. PostOrden
5. Calcular altura del árbol
6. Calcular grado del árbol
7. Imprimir árbol
8. Buscar un valor
9. Eliminar Nodo
10.Medir Tiempo
11.Salir

Opción: 7
Gráfica del árbol:
16
14
12
8
6
4
2

Seleccione una operación:
1. Agregar nodo
2. PreOrden
3. InOrden



Captura de Medición de tiempos