



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Estructuras de Datos
Taller 3: Árboles de Búsqueda, 2022-10

1 Objetivo

Evaluar la eficiencia de los árboles binarios ordenados en operaciones de búsqueda de elementos. En especial, se desea evaluar las habilidades del estudiante en el uso y análisis de las operaciones de inserción, eliminación y búsqueda de datos en árboles AVL.

2 Recordatorio: compilación con g++

La compilación con g++ (compilador estándar que será usado en este curso para evaluar y calificar las entregas) se realiza con los siguientes pasos:

1. **Compilación:** de todo el código fuente compilable (**ÚNICAMENTE LOS ARCHIVOS CON EXTENSIONES** *.c, *.cpp, *.cxx)

```
g++ -std=c++11 -c *.c *.cxx *.cpp
```

2. **Encadenamiento:** de todo el código de bajo nivel en el archivo ejecutable

```
g++ -std=c++11 -o nombre_de_mi_programa *.o
```

Nota: Estos dos pasos (compilación y encadenamiento) pueden abreviarse en un sólo comando:

```
g++ -std=c++11 -o nombre_de_mi_programa *.c *.cxx *.cpp
```

3. **Ejecución:** del programa ejecutable anteriormente generado
./nombre_de_mi_programa

ATENCIÓN: Los archivos de encabezados (*.h, *.hpp, *.hxx) **NO SE COMPILAN**, se incluyen en otros archivos (encabezados o código). Así mismo, los archivos de código fuente (*.c, *.cpp, *.cxx) **NO SE INCLUYEN**, se compilan. Si el programa entregado como respuesta a este taller no atiende estas recomendaciones, automáticamente se calificará la entrega sobre un 25% menos de la calificación máxima.

3 Desarrollo del taller

Sea x_1, x_2, \dots, x_n una muestra aleatoria de n observaciones ordenada de manera creciente. Para estos datos, la mediana (segundo cuartil o percentil 50) representa el valor de la variable ubicada en la posición central; si n es un número par, la posición de la mediana estará dada por la media aritmética de los dos valores centrales:

$$M = \begin{cases} X_{\frac{n}{2}} & \text{si } n \text{ es impar} \\ \frac{1}{2} (X_{\frac{n}{2}} + X_{\frac{n}{2}+1}) & \text{si } n \text{ es par} \end{cases}$$

El desarrollo del taller consistirá en utilizar su implementación de árbol AVL para organizar unos datos de entrada y así identificar la mediana de esos datos.

La información necesaria para poblar el árbol se encontrará en un archivo de texto que contendrá entre 10.000 y 100.000 registros. Cada registro inicia con un identificador: E (eliminar) o A (agregar); éstos hacen referencia a la operación que debe realizar sobre el árbol AVL con el número ubicado después del identificador. Después de realizar todas las operaciones, de adición y eliminación, el programa deberá identificar la mediana de los datos.

Las tareas puntuales que deberá realizar para el desarrollo de este taller son las siguientes:

1. Utilizar su implementación del árbol AVL, para cargar la información de los datos.
2. Dada su implementación, la información deberá quedar adecuadamente distribuida y organizada dentro del árbol.

3. Encontrar el nivel en el cual se encuentra la mediana y especificar su valor. Si el tamaño de la muestra es par, deberá encontrar el nivel de los dos datos centrales, obtener su valor y calcular la mediana.

La Figura 1 presenta un ejemplo del comportamiento de un árbol AVL a medida que se agregan nuevos nodos. En este ejemplo, al llegar a la figura (d), podemos decir que los datos centrales son 6 y 7, y que la mediana es 6.5.

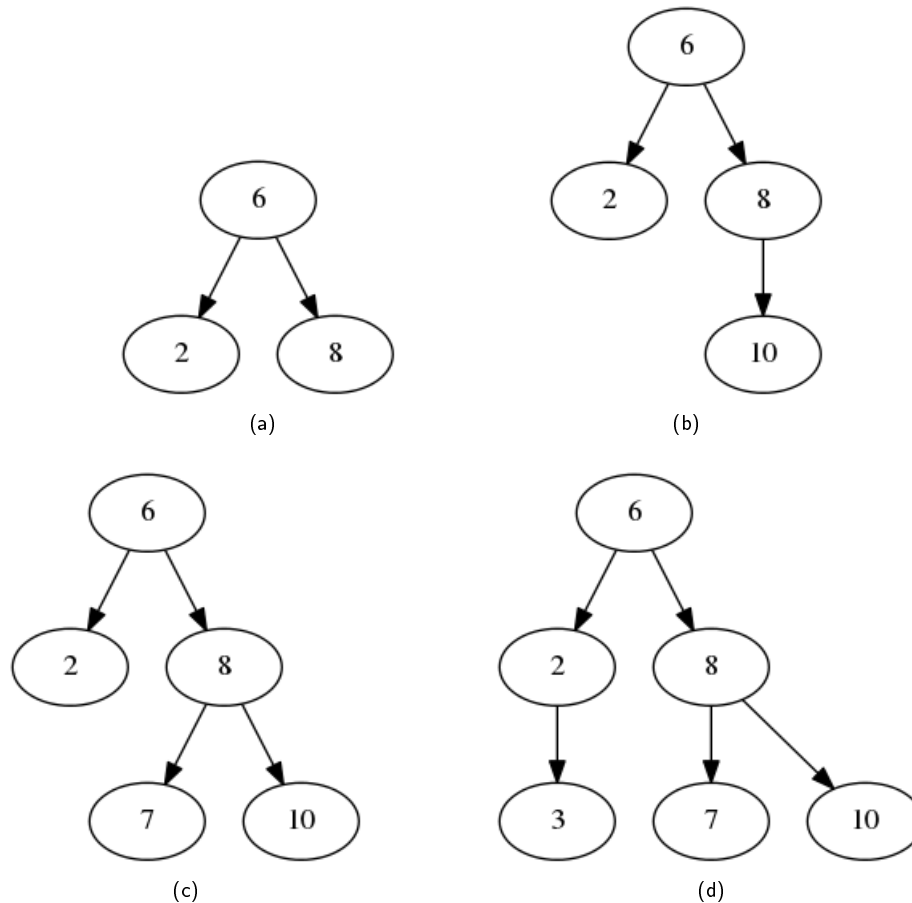


Figure 1: Ejemplo

4 Ejemplo de archivo de entrada

A continuación se presenta un ejemplo del archivo de entrada:

```
A 4 // agrega el número 4 al árbol
A 5
A 7
E 4 // elimina el número 4 del árbol
E 5
```

5 Evaluación

Como parte del desarrollo del taller se deberá entregar:

1. Documento de diseño.
2. Implementación propia del correspondiente TAD árbol AVL.
3. Código fuente funcional que realiza la carga del archivo en el árbol.

La entrega se hará a través de la correspondiente asignación de BrightSpace, antes de la medianoche del próximo domingo 20 de marzo de 2022. Se debe entregar un único archivo comprimido (únicos formatos aceptados: .zip, .tar, .tar.gz)

que contenga dentro de un mismo directorio (sin estructura de carpetas interna), documentos (único formato aceptado: .pdf) y código fuente (.h, .hxx, .cxx, .cpp). Si la entrega contiene archivos en cualquier otro formato, será descartada y no será evaluada, es decir, la nota definitiva de la entrega será de 0 (cero) sobre 5 (cinco).

La evaluación del taller tendrá la siguiente escala para cada una de las actividades o secciones de código a completar:

- **Excelente (5.0/5.0):** El programa se ejecuta adecuadamente, además fue bien diseñado y el código fuente refleja ese diseño.
- **Bueno (3.5/5.0):** El programa se ejecuta adecuadamente, pero el diseño es pobre y/o el código fuente no refleja el diseño.
- **No fue un trabajo formal de ingeniería (3.0/5.0):** El programa se ejecuta adecuadamente, pero no se diseñó.
- **Necesita mejoras sustanciales (2.0/5.0):** El programa no se ejecuta adecuadamente (no adiciona la información al árbol y/o encuentra la mediana de otra manera).
- **Malo (1.0/5.0) :** El código entregado por el estudiante no compila en el compilador g++ (mínimo versión número 4.5).
- **No entregó (0.0/5.0) :** No se entregan los archivos solicitados, o los archivos no se encuentran en el formato indicado.