



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Estructuras de Datos

Practica 07: Diccionario con árbol AVL

Prof. Edgardo Adrián Franco Martínez

<http://computacion.cs.cinvestav.mx/~efranco>

[@efranco_escom](#)

efranco.docencia@gmail.com





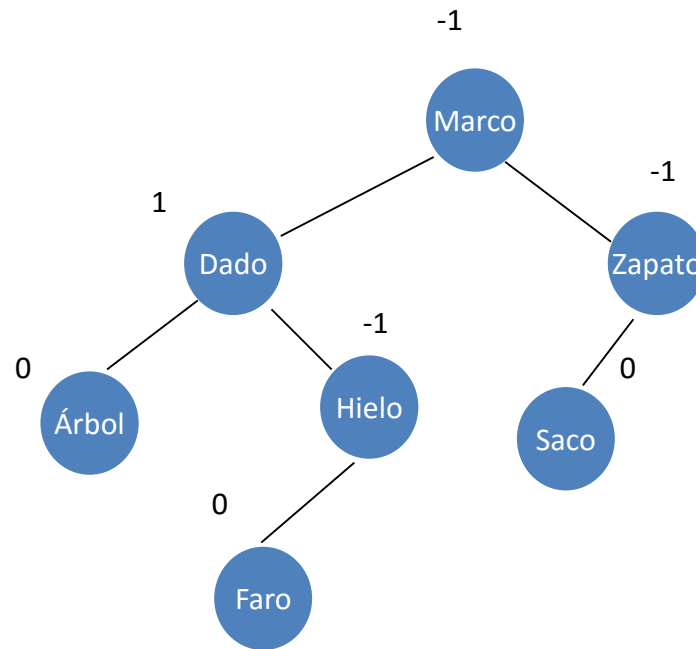
Contenido

- Problema
- Descripción
- Observaciones
- Opcional
- Requerimientos del reporte
- Fechas de entrega



Problema

- Con la implementación del TAD árbol binario de búsqueda y las reglas de un árbol balanceado AVL, construir un árbol capaz de almacenar palabras y sus definiciones (*Diccionario de palabras*).





- El árbol AVL, deberá de utilizar la función “**strcmp()**”, para distinguir entre una palabra mayor o menor a otra, para las operaciones de comparación en el ABB con su operación como AVL.

```
int strcmp(const char * s1,const char * s2);
```

- La función retorna un número entero mayor, igual, o menor que cero, apropiadamente según la cadena apuntada por s1 es mayor, igual, o menor que la cadena, s2);





Requerimientos

- El programa deberá mostrar un menú principal con las siguientes opciones:

Diccionario AVL

- 1.- Cargar un archivo de definiciones*
- 2.- Agregar una palabra y su definición*
- 3.- Buscar una palabra*
- 4.- Modificar una definición*
- 5.- Eliminar una palabra*
- 6.- Recorrido PreOrden*
- 7.- Recorrido InOrden*
- 8.- Recorrido PostOrden*
- 9.- Salir*





1.- Cargar un archivo de definiciones

- Realiza la carga en el árbol AVL de un archivo que contiene una palabra y su definición en cada línea del archivo.

2.- Agregar una palabra y su definición

- Permite agregar una palabra nueva al diccionario y su definición.

3.- Buscar una palabra

- Permite que se busque una palabra y consulte su definición

4.- Modificar una definición

- Permite que se busque una palabra y modifique su definición





5.- Eliminar una palabra

- Permite buscar una palabra y eliminarla.

6.- Recorrido PreOrden

- Muestra el recorrido PreOrden del árbol AVL

7.- Recorrido InOrden

- Muestra el recorrido InOrden del árbol AVL

8.- Recorrido PostOrden

- Muestra el recorrido PostOrden del árbol AVL

9.- Salir

- Sale del programa y elimina el árbol AVL.





- Se sugiere utilizar la estructura elemento siguiente para almacenar las palabras y sus definiciones.

```
typedef struct elemento
{
    char p[50]; //Palabras de tamaño 50
    char d[250]; //Definición de la palabra de no
    más de 250 caracteres
}elemento;
```

- Las palabras que provienen de un archivo vienen de la siguiente manera

Definición

Palabra(s):

Castaña / castañazo: Puñetazo.
Catanga: Persona de color
Cataplasma: Latoso, fastidioso
Baldosa / balero: Cabeza.
Balero: Cabeza
Banana, muy de la: Excelente.
Banca: Influencia / Banquero.
Bancar: pagar.



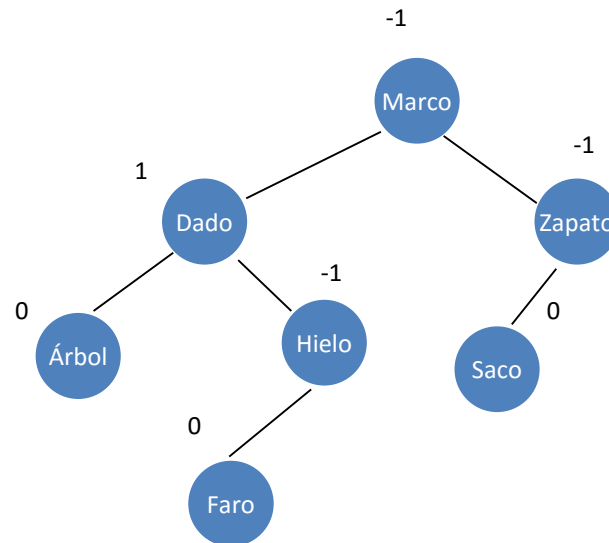
Observaciones

- Se deberá de realizar la implementación completa del árbol AVL, y mostrar las estadísticas del árbol, número de nodos, altura y en el caso de que se realicen balanceos mostrar cuál se hace en cada inserción o eliminación de palabras.
- El programa permite poder agregar nuevas palabras, consultar el significado de alguna, modificar el significado y eliminar palabras del diccionario AVL.
- Cuando se realizan búsquedas o consultas de palabras se deberá de mostrar el número de saltos que se hizo en el árbol para encontrarla.
- Se deberá de determinar si un nodo es mayor o menor que otro utilizando la función “strcmp()” de “string.h”.





- El programa deberá de ser capaz de cargar un archivo con palabras y sus definiciones, así como permitir al usuario saber el orden máximo de búsqueda, número de pasos al consultar una palabra, modificarla o eliminarla.
- Se deberá separar la implementación del árbol AVL, del árbol binario de búsqueda y del árbol binario.
- No puede haber dos definiciones para una misma palabra; i.e. no existen palabras repetidas.





Opcional

- Algunos puntos que da valor agregado a la calificación son:
 - El usuario puede exportar en determinado momento la lista de palabras a un archivo. (PreOrden, InOrden y PostOrden)
 - Se puede buscar todas las palabras que inicien con:
 - Una letra
 - Una frase
 - Que contengan una subcadena
 - Exportar una definición a un archivo.

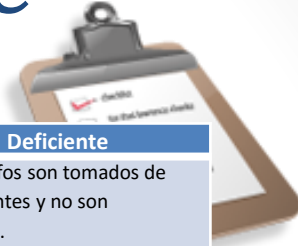




Reporte de practica

- Portada
- Introducción
- Planteamiento del problema
- Funcionamiento (Verificación de la solución, pruebas y resultados estadísticos de los balanceos del árbol AVL)
- Errores detectados (Si existe algún error detectado, el cuál no fue posible resolver o se desconoce el motivo y solo ocurre con ciertas condiciones es necesario describirlo)
- Posibles mejoras (Describir posibles disminuciones de código en la implementación o otras posibles soluciones)
- Conclusiones (Por cada integrante del equipo)
- Anexo (Códigos fuente *con colores e instrucciones de compilación)
- Bibliografía (En formato IEEE)





Rubrica de evaluación del reporte

Indicador	Excelente	Muy bien	Bien	Deficiente
Construcción de párrafos	Todos los párrafos incluyen una introducción, explicaciones o detalles y una conclusión	Los párrafos incluyen información relacionada pero no fueron generalmente bien organizados	La estructura del párrafo no estaba clara y las oraciones no estaban generalmente relacionadas	Los párrafos son tomados de otras fuentes y no son originales.
Redacción	No hay errores de gramática, ortografía y puntuación y la redacción es coherentemente	No hay errores de gramática, ortografía y puntuación, pero la redacción presenta incoherencias	Pocos errores de gramática, ortografía y puntuación	Muchos errores de gramática, ortografía y puntuación
Cantidad de información <small>Portada, Introducción, Planteamiento del problema, algoritmos e implementación, actividades y pruebas, errores detectados, posibles mejoras, conclusiones y anexos</small>	Todos los temas son tratados de manera clara y precisa, según lo solicitado.	La mayoría de los temas son tratados de manera clara y precisa	Dos temas no están tratados o están imprecisos y no cumplen lo solicitado.	Tres o más temas no están tratados o están imprecisos y no cumplen lo solicitado.
Calidad de la información	La información está claramente relacionada con el tema principal y proporciona varias ideas secundarias y/o ejemplos	La información da respuestas a las preguntas principales, y solo da algunos detalles y/o ejemplos	La información da respuestas a las preguntas principales, pero no da detalles y/o ejemplos	La información tiene poco o nada que ver con las preguntas planteadas.
Algoritmos	Los algoritmos dan solución apoyándose de pseudocódigo, diagramas y/o figuras en un lenguaje claro.	La mayoría de los algoritmos dan solución apoyándose de pseudocódigo, pero diagramas y/o figuras.	Los algoritmos son mencionados textualmente pero no se describen	Los algoritmos no son expresados en el reporte.
Organización	La información está muy bien organizada con párrafos bien redactados y con subtítulos con estilos adecuados	La información está organizada, pero no se distingue en estilos adecuados	La información está organizada, pero los párrafos no están bien redactados	La información proporcionada no parece estar organizada o es copiada de referencias externas de manera literal



Entrega vía Web



Grupo	Contraseña
1CM12	estructuras1cm12
1CM13	estructuras1cm13

- **En un solo archivo comprimido** (ZIP, RAR, TAR, JAR o GZIP)
 - Reporte (DOC, DOCX o PDF)
 - Códigos fuente (.C, .H, etc.)
 - **Código documentado:** Titulo, descripción, fecha, versión, autor.
 - *(Funciones y Algoritmos: ¿Qué hace?, ¿Cómo lo hace?, ¿Qué recibe?, ¿Qué devuelve?, ¿Causa de errores?).*
 - OBSERVACIONES
 - *NO enviar ejecutables o archivos innecesarios, las instrucciones de compilación van en el anexo del reporte. (Yo compilare los fuente)



Fechas de entrega



- **Demostración** *Laboratorio de Programación 1 (1107)*
- 1CM12, 1CM13 “Miércoles 06 de diciembre o miércoles 13 de diciembre de 2017”.



Entrega de reporte y código

- En un solo archivo comprimido.



Fecha y hora limite de entrega vía Web

- Martes 12 de Diciembre de 2017 a las 23:59:59 horas.

