Taller 03 – Documento de diseño

Estructuras de datos

23-10

Samuel Peña – Sebastián Córdoba-Sofia Galindo

Índice

Objetivo:

utilizar una 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.

Requerimientos

- Utilizar su implementación del árbol AVL, para cargar la información de los datos
- Dada su implementación, la información deberá quedar adecuadamente distribuida y organizada dentro del árbol.
- Encontrar el valor de la mediana. Si el tamaño de la muestra es par, deberá encontrar el valor de los dos datos centrales y calcular la mediana.

TAD´s

AVLNode

El TAD AVLNode representa los nodos que conforman el árbol AVL, los cuales tienen un valor numérico de tipo entero, y dos punteros a otros nodos: uno a su hijo izquierdo (left) y otro a su hijo derecho (right). También cuentan con una variable de altura (height) que se utiliza para calcular el factor de equilibrio y balancear el árbol.

- int data

- AVLNode* left

- AVLNode* right

- int height

+ AVLNode(int data)

+ AVLNode* minValueNode(AVLNode* node)

+ AVLNode* erase(AVLNode* node, int data)

+ AVLNode* insert(AVLNode* node, int data)

+ AVLNode* rightRotate(AVLNode* node)

+ AVLNode* leftRotate(AVLNode* node)

+ int balanceFactor(AVLNode* node)

+ int height(AVLNode* node)

+ void updateHeight(AVLNode* node)

+ void inorder(AVLNode* node)

AVLTree

AVLTree es el TAD que representa el árbol AVL en sí mismo, que tiene un puntero a la raíz del árbol (root), y que cuenta con los métodos necesarios para insertar, eliminar y recorrer los nodos del árbol.

- AVLNode* root

+ AVLTree()

+ AVLNode* insert(AVLNode* node, int data)

+ AVLNode* erase(AVLNode* node, int data)

+ void inorder(AVLNode* node)

+ int balanceFactor(AVLNode* node)

+ int height(AVLNode* node)

+ AVLNode* leftRotate(AVLNode* node)

+ AVLNode* rightRotate(AVLNode* node)

+ AVLNode* minValueNode(AVLNode* node)

+ void updateHeight(AVLNode* node)

sInfoEntrada

sInfoEntrada es una estructura que contiene información sobre los datos de entrada que se leen desde un archivo de texto. Tiene un carácter que indica el tipo de operación a realizar ('A' para agregar y 'E' para eliminar), y un valor numérico de tipo entero que representa el dato a agregar o eliminar del árbol AVL. Además, tiene dos métodos: leerlinea, que se utiliza para leer una línea del archivo y retornar una estructura sInfoEntrada, y dos métodos de clase (estáticos) que se utilizan para leer y filtrar la información de entrada desde un archivo de texto.

- char tipo

- int dato

+ sInfoEntrada leerlinea(char *linea)

+ vector<sInfoEntrada> infoEntradaAgregacion(string filename)

+ vector<sInfoEntrada> infoEntradaEliminacion(string filename)