Trabajo Práctico N°1

Licenciatura en Ciencias de la Computación

Estructuras de datos y algoritmos I

Cátedra: Federico Severino Guimpel, Mauro Lucci, Martín Ceresa, Emilio López, Valentina Bini

Implementación de árboles de intervalos con intérprete

Lucas Bachur, Tomás Scalbi

2020

Índice

1. Introducción

1.1. Motivación del trabajo

Los objetivos del trabajo se pueden dividir en 2, el primero es desarrollar una implementación para *árboles de intervalos* valiéndose de **árboles AVL**. Y luego la implementación de un intérprete para manipular este tipo de árboles desde la consola.

1.2. Flujo de trabajo

Trabajamos juntos en absolutamente todo. Nos organizamos para conectarnos a las 15:00 todos los días y avanzamos unas 3-4 horas por día en llamada a traves de Discord, el que escrbía el código compartía pantalla. Elegimos trabajar de esta forma porque no es un trabajo simple y cada pedazo necesita atención de ambos, y luego al final definiamos el objetivo para el día siguiente.

Al igual que en el trabajo anterior, lo hicimos a traves de github para tener un control de versiones seguro y práctico.
Fuimos alternando a lo largo del desarrollo entre Windows y Ubuntu, los 2 sistemas operativos con los que contamos para así ir probando que funcione en los 2 y no encontrarnos a último momento problemas de este estilo.

1.3. Breve descripción

Nos parecio interesante emplear tambien las estructuras de datos de pilas y colas por lo que optamos por hacer las funciones de AVLtree_recorrer_dfs y AVLtree_recorrer_bfs de forma iterativa. Elegimos la version de recorrer en pre-order porque nos resulto la más sencilla de pensar.
También nos resulto muy útil durante el desarrollo del trabajo la utilización de *paint*para visualizar el comportamiento esperado de nuestras funciones antes de programarlas.

2. Colas

2.1. Implementación

Al principio nos costó decidir a que llamariamos comienzo y a que final. No estabamos seguros si los nodos debían apuntar a su “anterior” o a su “siguiente”. Pero terminamos razonandolo según el ejemplo de una cola para un cajero. El comienzo de la cola es en el cajero, y la persona(nodo) que está enfrente, apunta hacia la persona que tiene detrás. El fin de la cola es donde se agregan las personas(nodos). Cada persona(nodo) tiene la información sobre quién está detrás, por lo tanto cuando se agrega una persona(nodo), el final actual de la cola(persona) debe ponerse al tanto que el lugar de atras dejo de estar vacio, y ahora se encuentra esta nueva persona(nodo).

3. Pilas

3.1. Implementación

La implementacion de pilas fue muy sencilla aunque también resultó un poco anti-intuitivo al comienzo la forma de desapilar y quitar elementos de la misma. Aunque la estructura base de la pila y cola que es:

```
1  typedef struct _GNodo {
2  void *dato;
3  struct _GNodo *next;
4  } GNodo;
5
```

Como tenían el mismo nombre, optamos por simplemente modificar el nombre de la estructura en el archivo de cabezera de la implementacion de pilas.

4. AVLTree

4.1. title

5. Bibliografía

Referencias

[1] **Geeksforgeeks** *Imprimir árbol binario en horizontal.*, Consultado en Mayo-Junio 2020
`https://www.geeksforgeeks.org/print-binary-tree-2-dimensions/`