

Matemáticas Aplicadas Y Ciencias de la Computación
/ MACC Entrega Final de Algoritmos y Estructura de
datos 2021-I

Arboles Rojo-Negro

Presentado por:

Natalia Katherine Rojas Suarez

Lenguaje de Programación de Desarrollo del Proyecto: C++ 14.

Repositorio: <https://github.com/Natalia-Rojas-Suarez/Algoritmos-rojo-negro>

Resumen ejecutivo:

Cuando deseamos balancear árboles binarios podemos encontrar distintas maneras de hacerlo, una de estas son los árboles rojo-negro, con estos árboles podemos asegurar que su altura va a mantenerse lo más baja posible, lo cual permitiría garantizar que se necesiten menos niveles en el mismo. También ayuda a que el tiempo de realización de las operaciones sea en el peor de los casos $\log n$.

Para realizar estos árboles tenemos que seguir unas propiedades muy específicas que los caracterizan: un nodo es rojo o negro no pueden ser ambos, la raíz siempre es negra, un nodo rojo siempre tendrá hijos negros, el número de nodos negros es el mismo en cualquier camino que vaya de la raíz a la hoja, todo camino de la raíz hasta las hojas debe tener el mínimo número de nodos negros, el camino más largo es dos veces el camino más corto y todo nodo que se inserta es rojo. Leo Guibas y Robert Sedgwick fueron quienes perfeccionaron e introdujeron la convención de colores.

Con el fin de comprender mejor cada una de estas propiedades y la funcionalidad de este mismo árbol, he creado este proyecto en donde podemos ver cómo se retira y cómo se agrega un nodo, cómo se reacomoda cada uno de los números que ingresan a este árbol y podemos comprender la funcionalidad de este. También podemos comprender cada una de las propiedades vistas anteriormente, ya que mostramos el color en el que se encuentra cada nodo y podemos notar en qué nivel se encuentra cada uno. La intención de este proyecto es ver de una manera más didáctica las propiedades que deben tener estos árboles para su óptima realización.

Funcionalidad de la herramienta

La funcionalidad de la herramienta consiste en crear un árbol rojo negro con las propiedades mencionadas anteriormente. Asegurando de que se obtenga un árbol con una baja altura desde la raíz, que se organicen adecuadamente los números y tengan los colores correspondientes. También desarrolle la eliminación de algún nodo al azar e igual garantizamos que se reorganice y se colocan los colores adecuados.

Así asegurándonos que cada una de estas partes funciona individualmente, al unir las cumplimos con las propiedades y aparte de esto podemos notar la manera en que se cumplen cada una de ellas. Además de esto, tenemos una interfaz simple de entender, la cual se puede reproducir desde la consola y es bastante gentil con el usuario a la hora de usarse. Para que así se pueda entender mejor el concepto de este árbol rojo-negro.

Algoritmos y estructuras de datos

Como mencionamos anteriormente, las partes más grandes de nuestro código consisten en balancear el árbol después de la eliminación, que la podemos encontrar en la parte privada de nuestra clase *ArbolNegroRojo*. En donde tenemos *eliminarArreglar*, que se encarga de balancear la eliminación que se hizo y que los colores se modifiquen correctamente; *nrTrasplantar*, esta posiciona la raíz; y por último tenemos *eliminar*, la cual se encarga; como lo dice su nombre; de quitar el nodo no deseado, aparte de que se asegura de que no se pueda eliminar algo que no existe en el árbol.

Otra de las partes mencionadas es balancear el árbol después de la inserción, dentro de este grupo tenemos, *balancear* que se encarga de acomodar los colores adecuados a cada uno de los nodos, e *imprimir* que se hace que, a la hora de ponerlo en nuestra consola, podemos ver claramente en qué nivel se encuentra cada nodo que tenemos en nuestro árbol.

Después tenemos el grupo de la inserción de un nodo, esta parte ya se encuentra en la parte publica de nuestro código. Tenemos *insertar*, que se encarga de agregar el nodo que desees; *eliminarNodo* que usa *eliminar* de la parte privada, e *imprimirArbol* que usa *imprimir* también de la parte privada; y éste, además, se encarga de que es imprima todo el árbol.

Por último, en el *main*, podemos encontrar un menú de opciones que son los que nos muestran en la consola para poder manejar visualmente nuestro árbol y poder ver cómo se implementan lo que hemos mencionado anteriormente. Agregado a esto, es la parte en donde se guía al usuario para que sepa que está realizando el programa.

Bibliografía

- https://es.wikipedia.org/wiki/Árbol_rojo-negro