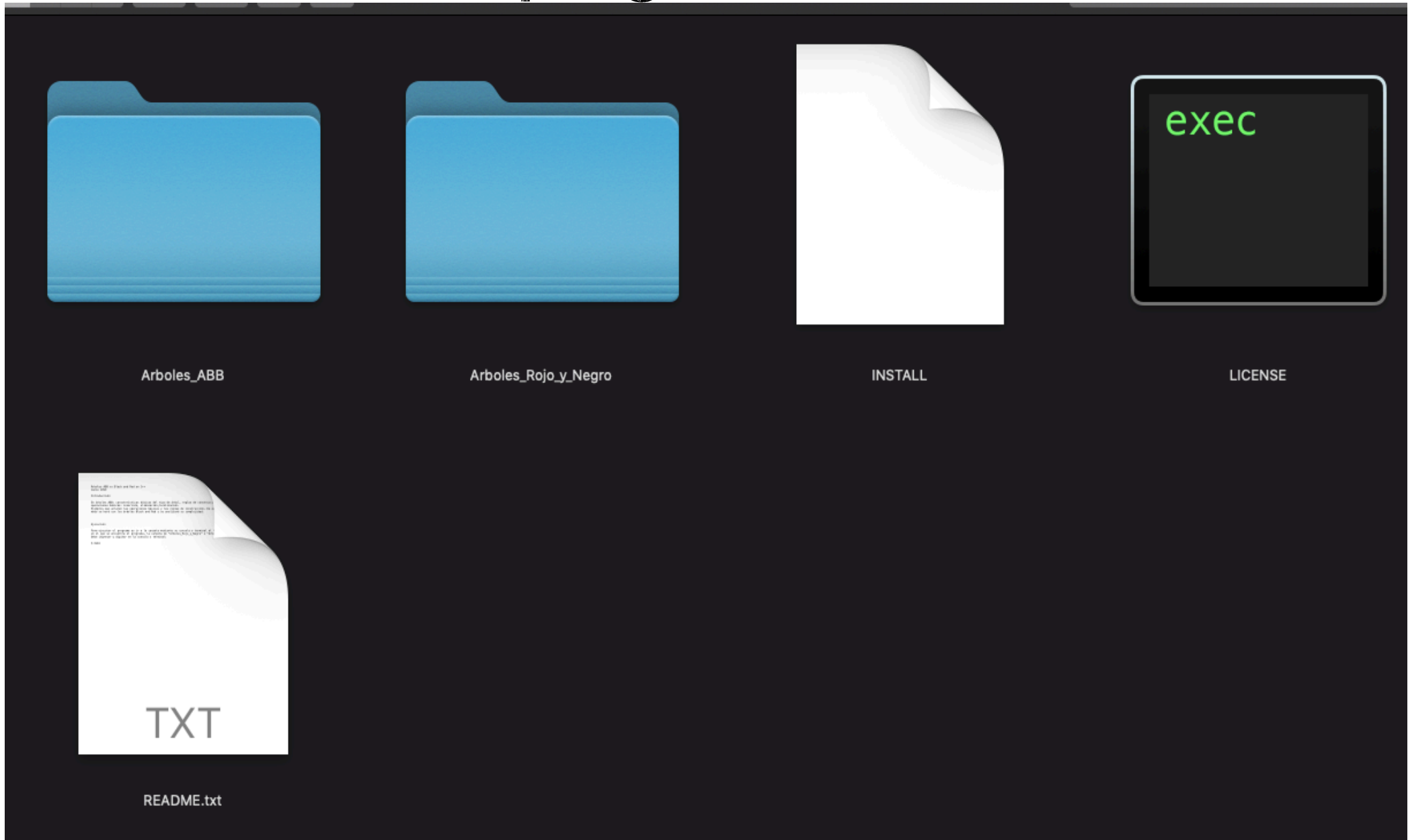


ABB vs Negro & Rojo



Funcionamiento del programa



NEGRO Y ROJO



arbol_rojo_negro.h



doxygen.pdf



header.h



main.cpp



makefile



ABB



ArbolABB.cpp



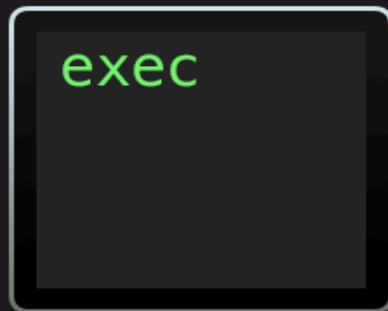
ArbolABB.h



doxygen.pdf



main.cpp



makefile



NodoArbol.cpp



NodoArbol.h



Readme

Árboles ABB vs Black and Red en C++
Junio 2019

Introducción:

En árboles ABB: características básicas del tipo de árbol, reglas de construcción, operaciones básicas: inserción, eliminación, localización. Ejemplos que aclaran las operaciones básicas y las reglas de construcción. De este mismo modo se hará con los árboles Black and Red y se analizará su complejidad.

Ejecución:

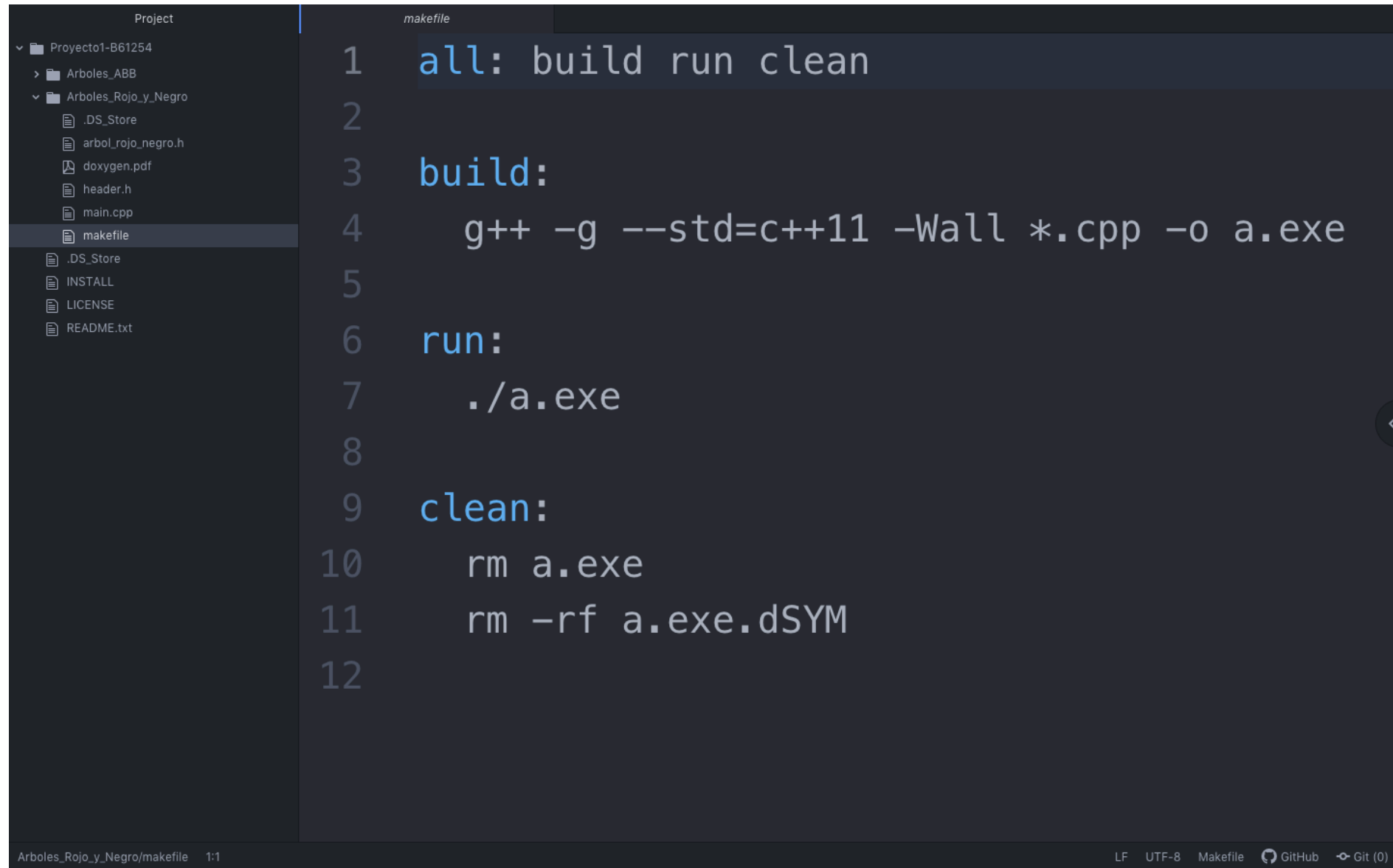
Para ejecutar el programa es ir a la carpeta mediante su consola o terminal al folder en el que se encuentra el programa, la carpeta de "Arboles_Rojo_y_Negro" o "Arboles_ABB". Debe ingresar y digitar en la consola o terminal:

```
$ make
```



Ejecutarlo

Makefile



The image shows a code editor with a dark theme. On the left is a file explorer showing a project structure. The main area displays a Makefile with four targets: 'all', 'build', 'run', and 'clean'. The 'all' target depends on 'build', 'run', and 'clean'. The 'build' target uses 'g++' with various flags to compile *.cpp files into a.exe. The 'run' target executes ./a.exe. The 'clean' target removes a.exe and a.exe.dSYM.

```
Project
├── Proyecto1-B61254
│   ├── Arboles_ABB
│   └── Arboles_Rojo_y_Negro
│       ├── .DS_Store
│       ├── arbol_rojo_negro.h
│       ├── doxygen.pdf
│       ├── header.h
│       ├── main.cpp
│       └── makefile
├── .DS_Store
├── INSTALL
├── LICENSE
└── README.txt
```

```
makefile
1  all: build run clean
2
3  build:
4      g++ -g --std=c++11 -Wall *.cpp -o a.exe
5
6  run:
7      ./a.exe
8
9  clean:
10     rm a.exe
11     rm -rf a.exe.dSYM
12
```

Arboles_Rojo_y_Negro/makefile 1:1

LF UTF-8 Makefile GitHub Git (0)

Header

```
header.h
1  #ifndef HEADER_H
2  #define HEADER_H
3
4
5
6  #include <algorithm>
7  #include <bitset>
8  #include <complex>
9  #include <deque>
10 #include <exception>
11 #include <fstream>
12 #include <functional>
13 #include <iomanip>
14 #include <ios>
15 #include <iosfwd>
16 #include <iostream>
17 #include <istream>
18 #include <iterator>
19 #include <limits>
20 #include <list>
21 #include <locale>
22 #include <map>
23 #include <memory>
24 #include <new>
```



```
25  #include <numeric>
26  #include <ostream>
27  #include <queue>
28  #include <set>
29  #include <sstream>
30  #include <stack>
31  #include <stdexcept>
32  #include <streambuf>
33  #include <string>
34  #include <typeinfo>
35  #include <utility>
36  #include <valarray>
37  #include <vector>
38  #include "arbol_rojo_negro.h"
39
40  #endif
41
```



Experimentos realizados

Tester de arboles ABB #1

Agregando valores

Agregar 35
Agregar 12
Agregar 50
Agregar 8
Agregar 17
Agregar 32
Agregar 53
Agregar 3
Agregar 10
Agregar 29
Agregar 57

Comprobando semejanza entre árboles

Como resultado obtenemos que:
No son semejantes

Prueba de isomorfismo entre árboles

Como resultado obtenemos que:
No son isomorfos

Prueba de espacio en los árboles

Realizando un recorrido a lo ancho del primer árbol
35 12 50 8 17 53 3 10 32 57 29

Realizando un recorrido a lo ancho del segundo árbol
53 12 57 8 50 59 41 24 80 3 10 17 32 29
Como resultado obtenemos que:
No esta lleno

Como resultado obtenemos que:
No esta completo

Eliminado hojas

Borrando hoja 29 1
Realizando un recorrido a lo ancho35 12 50 8 17 53 3 10 32 57
Borrando nodo con 1 hijo 53 1
Realizando un recorrido a lo ancho35 12 50 8 17 57 3 10 32
Borrando nodo con dos hijos 121
Realizando un recorrido a lo ancho35 8 50 3 17 57 10 32

Tester de arboles ABB #2

El arbol (29,11,28,18,31): No esta lleno
El arbol (29,11,28,18,31,8): No esta lleno
El arbol (29,11,28,18,31,8) No esta completo
El arbol (29,11,28,18,31,8,9,5,10,12) Como resultado obtenemos que:
No esta completo
Los arboles (50,46,12,23,48,53,57,68,80) y (28,29,9,7,11,32,12,90,17)
Como resultado obtenemos que:
No son isomorfos
Los arboles(50,46,12,23,48,53,57,68,80) y (53,80,23,57,46,12,50,68,48)
Como resultado obtenemos que:

```
Borrando nodo con 1 hijo 53 1
Realizando un recorrido a lo ancho35 12 50 8 17 57 3 10 32
Borrando nodo con dos hijos 121
Realizando un recorrido a lo ancho35 8 50 3 17 57 10 32
```

Tester de arboles ABB #2

```
El arbol (29,11,28,18,31): No esta lleno
El arbol (29,11,28,18,31,8): No esta lleno
El arbol (29,11,28,18,31,8) No esta completo
El arbol (29,11,28,18,31,8,9,5,10,12) Como resultado obtenemos que:
No esta completo
Los arboles (50,46,12,23,48,53,57,68,80) y (28,29,9,7,11,32,12,90,17)
Como resultado obtenemos que:
No son isomorfos
Los arboles(50,46,12,23,48,53,57,68,80) y (53,80,23,57,46,12,50,68,48)
Como resultado obtenemos que:
Si son semejantes
```

////////////////////////////////////



////////////////////////////////////

Vamos a realizar las siguientes operaciones:

////////////////////////////////////

3 10 8 17 32 35 50 57

Borrando hoja 32 1

3 10 8 17 35 50 57

Borrando hoja 29 0

3 10 8 17 35 50 57

Borrando hoja 28 0

3 10 8 17 35 50 57

Borrando hoja 7 0

3 10 8 17 35 50 57

Borrando hoja 12 0

3 10 8 17 35 50 57

Borrando hoja 90 0

3 10 8 17 35 50 57

Borrando hoja 17 1

3 10 8 35 50 57

Borrando hoja 9 0

3 10 8 35 50 57

Borrando hoja 11 0

3 10 8 35 50 57

rm a.exe

rm -rf a.exe.dSYM

Belindas-MacBook-Air:Arboles_ABB belindabrown\$ █



```
Belindas-MacBook-Air:Arboles_Rojo_y_Negro belindabrown$ make
g++ -g --std=c++11 -Wall *.cpp -o a.exe
./a.exe
```

```
*****
```

Árbol rojo y negro

```
*****
```

1. Agregar elemento al árbol
2. Buscar un elemento
3. PRE-ORDER
4. POST-ORDER
5. Eliminar un elemento del árbol
6. Salir

```
*****
```

Escoja una opción ➡➡➡➡ 1

Va a ser insertado ... 45

Se agregó el valor

➡➡➡➡Escoja una opción ➡➡➡➡ 1

Va a ser insertado ... 23

Se agregó el valor

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 45 67

Se agregó el valor

»»»»Escoja una opción »»»» Favor digite una opción dentro del menú

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 67

Se agregó el valor

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 2

Se agregó el valor




```
Va a ser insertado ... 3
-----
Se agregó el valor
-----

>>>>>Escoja una opción >>>>> 1

Va a ser insertado ... 1
-----
Se agregó el valor
-----

>>>>>Escoja una opción >>>>> 1

Va a ser insertado ... 1
-----
Se agregó el valor
-----

>>>>>Escoja una opción >>>>> 1

Va a ser insertado ... 4
-----
Se agregó el valor
-----

>>>>>Escoja una opción >>>>> 1

Va a ser insertado ... 78
-----
Se agregó el valor
-----
```



»»»»Escoja una opción »»»» 3

Pre-Order »»»» »»»» 45[Negro]»»»» 3[Rojo]»»»» 1[Negro]»»»» 1[Rojo]»»»» 2[Rojo]»»»» 23[Negro]»»»» 4[Rojo]»»»» 67[Rojo]»»»» 45[Rojo]»»»» 78[Rojo]

»»»»Escoja una opción »»»» 4

Post-Order »»»» »»»» 1[Rojo]»»»» 2[Rojo]»»»» 1[Negro]»»»» 4[Rojo]»»»» 23[Negro]»»»» 3[Rojo]»»»» 45[Rojo]»»»» 78[Rojo]»»»» 67[Rojo]»»»» 45[Negro]

»»»»Escoja una opción »»»» 2

»»»»Escoja una opción »»»» 3

Pre-Order »»»» »»»» 45[Negro]»»»» 3[Rojo]»»»» 2[Negro]»»»» 1[Rojo]»»»» 23[Negro]»»»» 4[Rojo]»»»» 67[Rojo]»»»» 45[Rojo]»»»» 78[Rojo]

»»»»Escoja una opción »»»» 4

Post-Order »»»» »»»» 1[Rojo]»»»» 2[Negro]»»»» 4[Rojo]»»»» 23[Negro]»»»» 3[Rojo]»»»» 45[Rojo]»»»» 78[Rojo]»»»» 67[Rojo]»»»» 45[Negro]

»»»»Escoja una opción »»»» 6

Salir »»»»



Conclusiones

- Un ABB tiene la característica de que todos los subelementos de un nodo a su izquierda están los menores y a su derecha los mayores almacenados. Por otro lado, la complejidad de los árboles negro y rojo es mayor, ya que es un árbol de búsqueda binaria de datos con asignación de colores.
- Poseen las siguientes características: Los nodos tienen un color propio ya sea rojo o negro. Aparte de todo lo que cumple un árbol BB se debe cumplir lo siguiente:

Nodo rojo o negro

Raíz negra

Hojas negras

Hijos de todo nodo rojo son negros

Cada camino tiene la misma cantidad de nodos negros y esto se denomina altura del árbol



Referencias bibliográficas

- [1] Kroah-Hartman G Corbet. J, Rubini. A. Linux Coding. O'Reilly books, 1998.
- [2] Computer Science Labs. Tecnology- commands. O'Reilly books, 2018.
- [3] Mark Summerfield. Programming in C++: A Complete Introduction to the C++ Language. Anaya Multimedia, 2009.
- [4] A. M. Turing. On computable numbers with an application to the Entscheidungs problem. Proceedings of the london mathematical society, 1997.

