



Tecnológico Nacional de México.
Instituto tecnológico de Nuevo
León.



Ingeniería en Sistemas Computacionales
Lenguajes y Autómatas II

“Reporte Árbol Binario”

Carlos Enrique Bernal Araujo

#15480004

Guadalupe, Nuevo León a 12 de Septiembre del 2018,

INTRODUCCIÓN

Un árbol binario es aquel que solo tiene dos descendientes directos. Los árboles se caracterizan por almacenar sus nodos en forma jerárquica y no en forma lineal como lo son otro tipo de estructuras. Las aplicaciones de los arboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual es posible tomar decisiones con dos opciones en distintos puntos.

Un árbol binario es una estructura de datos, y como todas las demás ya sean pilas o listas, sirve para organizar datos para facilitar su manipulación, ya sea el ingreso, borrado o búsqueda de estos.

Una de las ventajas de los arboles binarios es su parte de la búsqueda ya que como en muchos algoritmos relacionados a esto se necesita tener la información ordenada, y precisamente con los arboles la información entra de esta forma, y puede ser mostrada en sus diferentes tipos de recorridos ya sea en inorden, postorden o preorden.

DESCRIPCIÓN

Se realizó un árbol binario en el cual, como su nombre lo indica, cada nodo tenía dos hijos ya sean de números enteros, letras y/o operadores matemáticos.

El objetivo de este proyecto era que el usuario pueda insertar operaciones aritméticas desde el teclado o desde un archivo de texto, obteniendo como resultado el árbol de expresiones de dicha expresión.

Pseudocodigo:

teclear expresion

guardar expresion en lista EI

for i in rango(longitud de EI).

if (EI[i] es numero):

guardar elemento al final de la lista

elif(EI[i] es diferente de ')':

si la pila no esta basia

agregar al final de pila EI[i]

si esta basia

se evalua la prioridad de operador

si EI[i] es igual a '('

mientras que el tope de la pila sea diferente de '('

agrega el tope de la pila al final de EP

```
    si el tope de pila es '('
        elimina el tope de la pila
    si tope es diferente de -1
        agrega el tope de la pila al final de EP

for i in range(len(EP)):
    si EP es numero
        agregas a elp
    si no
        creas expresion
        si elp no esta basia
            agregas el tope de elp a la izquierda de expresion
        si elp no esta basia
            agregas el tope de elp a la derecha de expresion
        agregas la expresion al final de exp
    si elp esta basia
        si exp no esta basia
            se toma el tope de exp
            agregas el tope de exp a la derecha de expresion
            agregas la expresion al final de exp
            se agrega la nueva expresion a la izquierda de exp
imprimir exp
```

EXPERIMENTO

Se realizaron 10 repeticiones ingresando diferentes expresiones, fue tomado el tiempo de cada uno para así observar como el uso de operadores afectan durante la ejecución del programa.

Equipo que se usó:

Lenovo 80TJ

Ram: 8GB

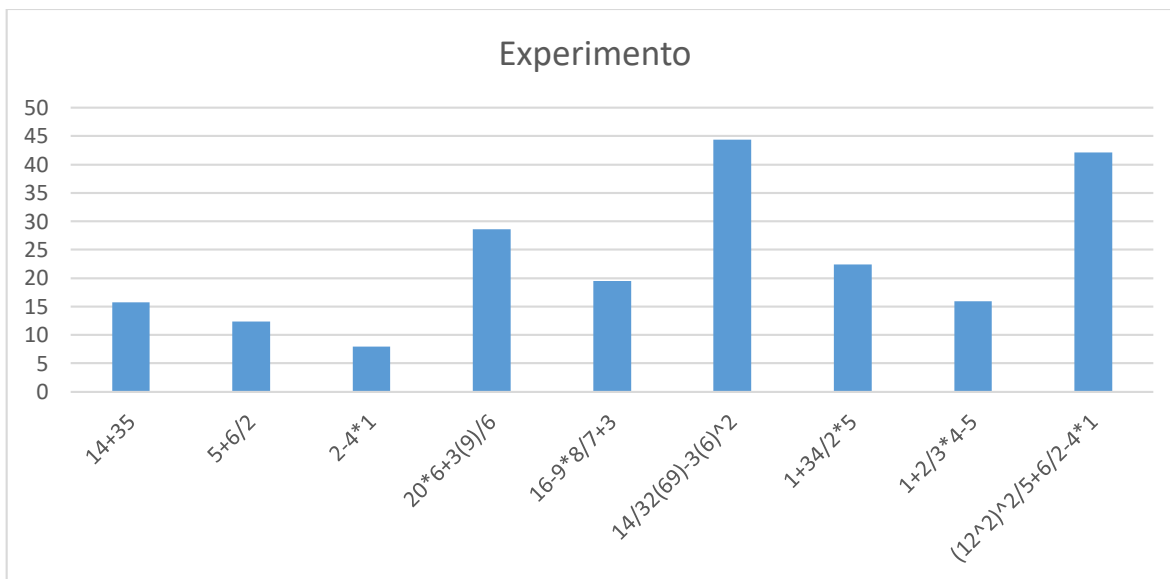
Procesador: AMD A8 con Amd Radeon
R5 Graphics a 2.20 GHz

Tipo de sistemas: 64 bits

RESULTADOS

Se ingresaron 10 expresiones diferentes al azar y se obtuvieron los siguientes tiempos como resultado.

Expresión	Tiempo (ms)
$14+35$	15.7702205
$2-4*1$	7.92615867
$5+6/2$	12.330924
$20*6+3(9)/6$	28.6365414
$16-9*8/7+3$	19.5277324
$14/32(69)-3(6)^2$	44.3293326
$1+34/2*5$	22.4457538
$1+2/3*4-5$	15.9180803
$(2^2)^2+3*9/8-2$	24.626663208007812
$(12^2)^2/5+6/2-4*1$	42.1585171



CONCLUSIONES:

Con este proyecto se creó un software capaz de leer una función entera, utilizando como base números y operadores aritméticos, y que esta sea mostrada con su respectivo árbol de expresión.

Al igual se utilizó la función time para medir los tiempos de ejecución de las diferentes expresiones.

Fue un proyecto interesante ya que utilizamos un lenguaje que, hasta el momento, no habíamos manejado totalmente como lo es Python.

BIBLIOGRAFÍA:

<https://www.foro.lospillaos.es/separar-cadena-y-hacer-el-arbol-de-una-ecuacion-vt11948.html>