Idea y Grupo para el Trabajo Práctico Especial

`v0.1.0`

Grupo

Nombre	Apellido	Legajo	E-mail
Marcos	Dedeu	60469	mdedeu@itba.edu.ar
Roberto Franco	Rodríguez	60089	robrodriguez@itba.edu.ar
Federico	Rojas	60239	frojas@itba.edu.ar
Santiago	Sandrini	61447	ssandrini@itba.edu.ar

Repositorio

- mdedeu/TP-TLA (github.com)

Idea

Consiste en un lenguaje específico que permite trabajar con árboles binarios, no binarios, completos, incompletos y balanceados. Se podrían aplicar operaciones cómo búsquedas, transformaciones, agregado de nodos, balanceos con diferentes técnicas. Se debería poder calcular la altura, el ancho, la cantidad de nodos, y otras variables relacionadas.

Prestaciones

1. Se podrán crear árboles binarios, no binarios, red-black trees, árboles B, AVL, y BST partiendo de un nodo inicial.
2. Se podrán crear nodos e insertarlos en los árboles.
3. Se podrán eliminar o modificar nodos.
4. Se podrá realizar búsquedas con diferentes algoritmos.
5. Las variables podrán ser del tipo Nodo, Árbol, ints o strings.
6. Las variables podrán ser vectores de alguno de los tipos anteriores.
7. Se proveerán operadores relacionales como <, >, =, ≠, ≤ y ≥.
8. Se proveerán operaciones aritméticas básicas como +, -, * y /.
9. Se proveerán operaciones lógicas básicas como AND, OR y NOT.
10. Se proveerán estructuras de control básicas de tipo IF-THEN-ELSE, FOR y WHILE.
11. Se podrán crear árboles a partir de otros árboles.
12. Se podrán graficar los árboles.

Ejemplo

Un ejemplo podría ser un programa en el cual se crea un árbol no binario que almacena nombres de personas dentro de una empresa:

```
main

tree departamentoIT ;

//se crea el árbol
departamentoIT = newTree();

//se crea el nodo raíz (data, id)
departamentoIT.createNode("Federico-CTO", "Federico-CTO");

//se agrega un nodo hijo (data, id, padre)
departamentoIT.createNode("Franco-VP", "Franco-VP", parent="Federico-CTO");

departamentoIT .createNode("Gonzalo-VP", "Gonzalo-VP", parent="Federico-CTO");

juniors[3] = {"Santiago-JN", "Marcos-JN", "Agustín-JN"};

for(int i = 0; i < juniors.length(); i++)

departamentoIT .createNode(juniors[i], juniors[i], parent="Franco-VP");

end

departamentoIT .removeNode("Gonzalo-VP");

node CTO = departamentoIT.searchKey("Federico-CTO");

CTO.print();

departamento.height.print();

departamentoIT.print();

return
```