

FUNCTIONS' COMPLEXITY

- **Complexity in the worst case of insert:**

$$T(n) = C1 + C2 + n$$

$$T(n) = n \quad \text{---> Sum law}$$

$O(n)$ where n is the size of the binary tree

- **Complexity in the worst case of search:**

$$T(n) = C1 + C2 + C3 + C4 + C5 + n*C6 + n*C7$$

$$T(n) = (C6+C7)*n \quad \text{---> Sum law and common factor}$$

$$T(n) = n \quad \text{---> Product law}$$

$O(n)$ where n is the size of the binary tree

- **Complexity in the worst case of delete an element:**

$$T(n) = C + C3*n + C5*n + *C13$$

$$T(n) = (C3+C5+C13)*n \quad \text{---> Sum law and common factor}$$

$$T(n) = n \quad \text{---> Product law}$$

$O(n)$ where n is the size of the binary tree

- **Complexity in the worst case of print:**

$$T(n) = C1 + C2 + n*C3 + n*C4$$

$$T(n) = (C3+C4)*n \quad \text{---> Sum law and common factor}$$

$$T(n) = n \quad \text{---> Product law}$$

$O(n)$ where n is the size of the binary tree