

# lab04 - Taller: TAD Diccionario

Algoritmos y Estructuras de Datos II 2020

## Preliminares

En este proyecto deberán implementar el TAD *Diccionario* en el lenguaje de programación C. Deben basarse en la solución elaborada para el teórico / práctico, intentando que sea una simple traducción de un lenguaje a otro. Dadas las diferencias entre C y el lenguaje utilizado en el práctico es inevitable realizar algunos cambios.

En C las declaraciones de las funciones del TAD Diccionario deben estar en el archivo dict.h, donde es necesario además definir el tipo dict. La implementación de estas funciones estarán en dict.c.

En el ejercicio práctico, la definición de dict se hace de la siguiente manera:

```
type Node of (K,V) = tuple
    left: pointer to (Node of (K,V))
    key: K
    value: V
    right: pointer to (Node of (K,V))
end tuple

type Dict of (K,V)= pointer to (Node of (K,V))
```

En C, lo mismo sería:

```
struct _node_t {
    dict_t left;
    dict_t right;
    key_t key;
    value_t value;
};

typedef struct _node_t *dict_t;
```

### **No olvidar!!!!**

Como invariante de representación debemos asegurar que el árbol representado por la estructura sea binario de búsqueda de manera que la operación de buscar un valor tenga orden logarítmico. Es decir, dado un nodo  $n$ , toda clave ubicada en el nodo de la derecha  $n.right$ , debe ser mayor o igual a  $n.key$ . Y toda clave ubicada en el nodo de la izquierda  $n.left$ , debe ser menor a  $n.key$ . Debes tener especial cuidado en la operación que agrega pares al diccionario.

## **Actividades**

Los archivos involucrados en este proyecto son:

**main.c** : el programa principal es un interfaz para poder cargar diccionarios, agregar y eliminar palabras, etc. Leer detenidamente y entender el código que entrega la cátedra.

**dict.c** : contiene una implementación incompleta del TAD Diccionario. Incluye la estructura elegida para la implementación. Se debe completar la implementación.

**dict.h** : contiene las definiciones necesarias del TAD Diccionario.

**key\_value.h** : contiene algunas definiciones de los tipos key y value.

**string.c** : contiene una implementación del TAD String, que se usa para las palabras y definiciones. Abrir el archivo y estudiarlo.

**string.h**: contiene las definiciones del tipo string.c.

### **Ejercicio 1**

Leer y entender el código entregado por la cátedra

### **Ejercicio 2**

Implementar todo las funciones entregadas en dict.c

