

Ejercitación – Hashing

Este ejercicio consiste en implementar un diccionario `DiccHash<V>` con claves de tipo `string` y valores de tipo `V`, utilizando como estructura una tabla de hash con direccionamiento abierto: cada *bucket* de la tabla guarda una lista de elementos. La representación ya se encuentra dada en `src/DiccHash.h`.

El proyecto posee un único target: `correrTests`. Puede ejecutarse desde CLion o utilizando `cmake` y `makefile` como de costumbre.

Completar la implementación de los siguientes métodos de la clase `DiccHash<V>` en el archivo `src/DiccHash.hpp`:

1. `DiccHash()` — Constructor sin parámetros de la clase.
2. `~DiccHash()` — Destructor.
3. `bool definido(const string& clav) const` — Devuelve `true` si la clave está definida en el diccionario.
4. `void definir(const string& clav, const V& sig)` — Agrega una clave al diccionario, con el significado dado. Si la clave a agregar ya se encuentra definida, redefine su significado.

Nota de implementación: si el factor de carga supera el valor de `UMBRAL_FC`, se debe redimensionar la tabla al doble de tamaño.

Pasos a seguir para redimensionar la tabla:

- (a) Crear una tabla del doble de tamaño de la original.
 - (b) Insertar todas las claves de la tabla original en la tabla nueva.
 - (c) Liberar la memoria reservada para la tabla original.
5. `V& significado(const string& clav)` — Devuelve el significado de la clave indicada. *Precondición:* la clave debe estar definida.
 6. `void borrar(const string& clav)` — Borra la clave indicada del diccionario. *Precondición:* la clave debe estar estar definida.
 7. `unsigned int cantClaves() const` — Devuelve el número de claves definidas en el diccionario.
 8. `set<string> claves() const` Devuelve el conjunto de claves del diccionario (representado como un `set<string>` de la STL de C++).

Además, se sugiere implementar la función de hash como un método privado:

9. `unsigned int fn_hash(const string& str) const` — Función de hash.