Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna

Grado en Ingeniería Informática
Algoritmos y Estructuras de Datos Avanzadas

Curso 2020-2021

Práctica 5: Tabla Hash con dispersión abierta

Objetivo

El objetivo de esta práctica es trabajar con los algoritmos de búsqueda y realizar la implementación en lenguaje C++ de la técnica de hashing con dispersión abierta. Se utilizará la definición de tipos genéricos, el polimorfismo dinámico y la sobrecarga de operadores.

Entrega

La entrega se realizará en la sesión de laboratorio entre el 5 y el 9 de abril. Durante esta sesión se podrán solicitar modificaciones sobre el enunciado de la práctica.

En el actual escenario de presencialidad adaptada, la entrega se realizará en tareas separadas para la rotación con asistencia presencial y la rotación con asistencia online. Si la ULL cambiase al escenario 2, docencia online, se comunicará la forma de entrega en el aula virtual.

Enunciado

Se pide desarrollar un tipo de dato genérico `TablaHash<class Clave>` para almacenar valores de tipo `Clave` utilizando la técnica de búsqueda por dispersión abierta. Esta técnica se basa en la existencia de funciones de dispersión, que reciben como parámetro un valor de tipo `Clave` y retorna la posición de la tabla, en el rango `[0..nDatos-1]`, donde corresponde almacenar dicho valor de clave. En la práctica se pide implementar, al menos, las funciones de dispersión ***módulo*** y ***pseudoaleatoria***.

Para definir de forma general las distintas funciones de dispersión, y permitir que en tiempo de ejecución se pueda elegir una función de dispersión particular, se encapsula la definición de la función de dispersión dentro de la clase genérica abstracta `FuncionDispersion<Clave>` que sobrecarga al operador función como un método nulo.

```
template<class Clave>
class FuncionDispersion {
  public:
    unsigned operator()(const Clave& k) const = 0;
};
```

De esta forma, la implementación de cada función de dispersión particular se realiza en el operador función de una clase derivada de la clase base abstracta. Por ejemplo, la función de dispersión módulo:

```
template<class Clave>
class fdModulo: public FuncionDispersion<Clave> {
  public:
    fdModulo(const unsigned n): nDatos(n){}
    unsigned operator()(const Clave& k) const {
      return k % nDatos;
    }
  private:
    unsigned nDatos;
 };
```

La clase genérica `TablaHash<Clave>` dispondrá de los siguientes miembros:

- Atributo privado `nDatos`, contiene el tamaño de la tabla. Se especifica en el constructor.
- Atributo privado `vDatos`, es un array de `nDatos` posiciones en cada una de las cuales habrá un contenedor de claves.
- Atributo privado `fd`, es un puntero a la clase base `FuncionDispersion<Clave>` que apunta a un objeto instanciado en tiempo de ejecución para alguna de las clases derivadas de la clase base.
- Método `Constructor`, recibe como parámetros el tamaño de la tabla y un objeto de tipo función de dispersión que se enlazará al puntero `fd`.
- Método `bool Buscar(Clave& X) const`, método público que retorna el valor booleano `true` si el valor `X` del tipo `Clave` está guardado en la tabla hash. En otro caso retorna el valor booleano `false`.
- Método `bool Insertar(const Clave& X)`, retorna el valor booleano `true` si puede insertar el valor `X` del tipo `Clave` en la tabla hash. En otro caso, si el valor ya está guardado en la tabla, se retorna el valor booleano `false`.

En la técnica de búsqueda con dispersión abierta cada posición de la tabla hash contiene una lista donde se insertan los ***sinónimos***, valores de clave distintos a los que les corresponde la misma posición en la tabla. Así, la inserción consiste en calcular la posición en la tabla asignada por la función de dispersión y agregar el nuevo registro a la lista correspondiente. Y de forma análoga, la búsqueda consiste en calcular la posición en la tabla asignada por la función de dispersión y buscar el registro en la lista correspondiente.

Se implementará una clase genérica `Lista<Clave>` para instanciar los contenedor de claves ubicados en cada posición de la tabla. Esta clase genérica dispondrá de los siguientes métodos:

- Método `Constructor`, inicializa un objeto lista vacía.
- Método `bool Buscar(Clave& X) const`, método público que retorna el valor booleano `true` si el valor `X` del tipo `Clave` está guardado en la lista. En otro caso retorna el valor `false`.
- Método `bool Insertar(const Clave& X)`, retorna el valor booleano `true` si puede insertar el valor `X` del tipo `Clave` en la lista. En otro caso, si el valor ya está guardado en la tabla, se retorna el valor booleano `false`.

El programa principal pide al usuario el tamaño de la tabla y la función de dispersión a utilizar. Crea el objeto que implementa la función de dispersión e instancia la tabla hash con el tipo de `Clave=int`. A continuación implementa un menú que permita insertar y buscar valores de clave solicitadas al usuario, mostrando el resultado de la operación.