TABLAS HASH

Maria C. Torres Madroñero
Profesora asociada
Departamento de Ciencias de la Computación y la Información

Objetivos

- Implementar la estructura de tablas hash y sus operaciones básicas
- Implementar una aplicación de tablas hash

Recursos requeridos

- PC
- IDE para JAVA o Python – el estudiante deberá seleccionar un lenguaje de programación para el desarrollo de las prácticas de laboratorio

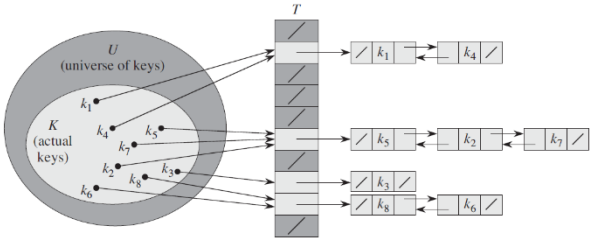
Actividades preliminares al laboratorio

- Lectura de la guía

Marco teórico

TABLAS HASH

También conocidas como tablas de dispersión, son una estructura de datos (permite manejar colecciones de objetos) que de forma efectiva buscan implementar las operaciones de insertar, eliminar y buscar de un diccionario. Un diccionario asocia a cada dato de la colección una clave usualmente numérica.



Las tablas hash son una forma de solucionar los problemas de memoria cuando un diccionario es implementado por direccionamiento directo. En el direccionamiento directo, un elemento con clave k se almacena en la posición k de un arreglo. En este sentido, es necesario un arreglo de tamaño igual al máximo número posible de las claves. Para evitar el uso ineficiente de memoria, las tablas de dispersión o hash emplean una función de dispersión, mapeando el universo de claves a posiciones de un arreglo de tamaño m, donde m es usualmente mucho más pequeño que el tamaño de k. Sin embargo, esta asignación puede causar problemas cuando dos claves se asignan a la misma posición de la tabla, esta situación se denomina una colisión. Existen diferentes mecanimso para solucionar colisiones, entre las mas usadas se encuentra la solución por encadenamiento (chaining) y direccionamiento abierto (open addressing).

Una tabla hash basada en encadenamiento consiste en ubicar todos los elementos que se direccionan a la misma posición dentro de una lista enlazada. Esto quiere decir que la tabla Hash es representada por un arreglo, y cada elemento del arreglo continue una lista enlazada, que se inicialmente esta vacía, pero en la cual se van encadenando los elementos que van colisionando. Para que los datos queden adecuadamente distribuidos a lo largo de las posiciones del arreglo se emplean funciones de dispersión. Una función de dispersión se espera que asigne una clave con igual probabilidad a las m posiciones del arreglo. Entre las funciones de dispersión más empleadas y simples se encuentra el método de la división:

$$h(k) = k \bmod m$$

En la ecuación anterior $h(k)$ es la función hash, k la clave, y m el tamaño de la tabla. Cuando se usa este método usualmente se evita que m sea una potencia de 2. Usualmente, un numero primero no muy cercano a una potencia de 2 es una buena elección.

Por otro lado, el método de la multiplicación primero escala la clave k por una contante A que se encuentra en el rango e (0,1) y se extrae la parte decimal. Este valor es multiplicado por m y se toma la parte entera:

$$h(k)=\lfloor m(kA \bmod 1)\rfloor$$

Este método trabaja para cualquier valor de A, sin embargo trabaja mejor para algunos valores dependiendo de los datos. Un valor sugerido de A es:

$$A\approx(\sqrt{5}-1)/2 = 0{,}6180339887\ldots.$$

Entre las principales aplicaciones de las tablas hash se encuentran la criptografía, el manejo de archivos de un sistema, verificación de contraseñas, búsqueda de patrones en cadenas, entre otros.

Actividades

Problema 1 Implementación de las clases Chained_Hash

En esta primera etapa del laboratorio, implementaremos la clase Chained_Hash. Usaremos un arreglo de tamaño m, y cada posición del arreglo almacenara una lista doble (usar la implementación de lista doble del laboratorio 4). Se sugiere que la implementación permita almacenar datos genéricos (tipo Object) asociados a una clave de tipo entero. El tamaño del arreglo se debe establecer como un parámetro del constructor. Así mismo, la clase Chained_Hash debe permitir elegir entre el método de la división o el de la multiplicación como función Hash (esta selección debe especificarse desde el constructor de la clase). La implementación de Chained_Hash debe permitir al menos realizar las siguientes operaciones:

- Insertar un nuevo dato: dado un dato E y su clave k, se determina con una función Hash la posición i del arreglo donde debe agregarse el objeto, y se añade a la respectiva lista usando el método addLast() de la lista doble.
- Buscar un dato dada su clave: se determina con la función Hash la posición i del arreglo donde debe encontrarse el objeto con la clave dada; se recorre la lista doble para encontrar el nodo, si se encuentra se retorna el objeto.
- Eliminar un dato dada su clave: se busca el nodo respectivo donde se encuentra el objeto (usando la posición del arreglo de acuerdo con la función hash) con la clave dada (note que el método buscar descrito anteriormente retorna el objeto no el nodo), se emplea el método remove() de la lista doble para eliminar el dato de la lista.
- Implemente dos funciones Hash: el método de la división y el método de la multiplicación. La función hash debe incluirse como un parámetro del constructor de la clase.

Problema 2 Prueba de la implementación

Vamos a suponer que vamos a crear una tabla Hash basada en un arreglo de 10 elementos. Genere 20 números enteros aleatorios desde 0 a 100. Para esta primera prueba el data E y la clave k corresponderá a los números aleatorios.

a. Inserte cada número aleatorio a la tabla Hash usando el método de la división
b. Busque uno de los números generados
c. Elimine uno de los número generados
d. Repita los pasos a-c empleando el método de la multiplicación

Problema 3 Prueba de la implementación

Ahora vamos a crear una tabla Hash con 5 posiciones, y vamos a insertar 6 usuarios (clase usuario del laboratorio 2). La clave en este caso corresponderá a los números de identificación de los usuarios.

a. Ingrese los usuarios a la tabla Hash empleando el método de la división, imprima en pantalla cuantos usuarios fueron almacenados en cada posición del arreglo
b. Ingrese los usuarios a la tabla Hash empleando el método de la multiplicación, imprima en pantalla cuantos usuarios fueron almacenados en cada posición del arreglo

Instrucciones de entrega

- La solución de los problemas debe desarrollarse en JAVA o Python. Los estudiantes tendrán la libertad de seleccionar el lenguaje de programación y plataforma para presentar la solución de los problemas.
- La solución debe emplear librerías nativas y se invita a los estudiantes a no usar código descargado de internet. Los laboratorios están diseñados para practicar los fundamentos teóricos; entre más código escriba el estudiante más fácil será su comprensión de los temas de clase.
- La solución se puede presentar en grupos de hasta 3 estudiantes.
- La solución de los problemas debe entregarse y sustentarse en el aula de clase o en la hora de asesoría a estudiantes. No se reciben soluciones por correo electrónico.