Tecnológico de Monterrey
Escuela de Ingeniería y Ciencias

Campus Sonora Norte

Reflexión Actividad 5.2

Curso:

Programación de estructuras de datos y algoritmos fundamentales

(TC1031)

Estudiante:

Daniel Alfredo Barreras Meraz A01254805

Docente:

Baldomero Olvera Villanueva

Fecha de entrega:

27 de noviembre de 2023

Las tablas de hash son estructuras de datos que se caracterizan por el uso de llaves o claves que almacenan valores ligados a ellas. Por ello, una de las principales funciones que se aplican en las tablas de hash es la “búsqueda”. Similarmente a como lo haríamos en una base de datos, esta función permite acceder a los elementos a partir de una clave generada usando por ejemplo: un nombre o un número de cuenta. Lo que se hace es transformar la clave en un número que la tabla utiliza para encontrar el valor que se necesita. De esta manera, en una tabla hash, los datos se almacenan en conjuntos de pares “clave - valor”.

La función hash es el corazón de una tabla de hash. Esta función toma una clave y la transforma en un número, que luego se utiliza como índice para acceder a la cubeta correspondiente en la tabla de hash. Aunque puede parecer sencillo, el diseño de una buena función hash es un desafío, ya que debe distribuir las claves de manera uniforme a través de la tabla de hash para minimizar las colisiones. Las colisiones ocurren cuando más de una clave se asigna a la misma cubeta. Aunque pueden parecer problemáticas, las colisiones son manejables. Una estrategia común es el encadenamiento, que implica crear una lista vinculada de valores en cada cubeta.

Las tablas de hash son especialmente útiles cuando se manejan grandes cantidades de información. Aunque pueden parecer similares a los arreglos en su implementación, las tablas de hash son más eficientes que los árboles de búsqueda u otras estructuras de búsqueda de tablas en muchas situaciones.

En términos de aplicaciones prácticas, las tablas de hash son omnipresentes en la informática. Se utilizan en muchos tipos de software, desde la implementación de matrices asociativas y la indexación de bases de datos hasta la creación de cachés y conjuntos. Un ejemplo cotidiano de una aplicación de tablas de hash es un diccionario, donde una clave puede ser representada por la palabra a buscar y el valor sería el significado de esta palabra.

Un ejemplo ilustrativo de la aplicación de las tablas de hash se presentó al abordar un desafío en la compañía naviera “International Seas, LTD.”. Esta empresa tenía la necesidad de calcular los salarios de sus empleados basándose en las descripciones de sus puestos de trabajo. Para resolver este problema, recurrí a las tablas de hash. En este caso, las palabras en la descripción del trabajo servían como claves. El primer paso para resolver el problema fue llenar la tabla de hash con las palabras del diccionario proporcionado y sus valores asociados. Esto se logró utilizando el método insertar de la clase HashTable.

Una vez que la tabla de hash estuvo llena, el siguiente paso fue calcular el salario para cada descripción de trabajo. Para ello, utilicé el método calcularSalario de la clase HashTable. Este método toma una descripción de trabajo, la divide en palabras y busca cada palabra en la tabla de hash. Si la palabra se encuentra en la tabla de hash, se suma su valor asociado al salario total.

La eficiencia de las tablas de hash fue crucial para resolver este problema de manera efectiva. A pesar de tener que buscar muchas palabras en la tabla de hash, el tiempo que tardó el sistema en calcular el salario fue relativamente corto. Esto se debe a que las tablas de hash permiten un acceso casi instantáneo a los elementos almacenados, lo que las hace ideales para tareas como esta.

En términos de complejidad algorítmica, la operación de inserción en una tabla de hash tiene una complejidad de tiempo promedio de O(1), mientras que la operación de búsqueda también tiene una complejidad de tiempo promedio de O(1). El método de hashing utilizado en la función hash es crucial para minimizar la probabilidad de este peor caso. En la función hash, se utiliza un método de hashing comúnmente conocido como hashing de multiplicación y adición. Este método toma cada carácter de la cadena de entrada, lo multiplica por un número primo (en este caso, 31), y luego suma el valor ASCII del carácter.

El resultado se toma módulo M, donde M es el tamaño de la tabla hash, para asegurar que el valor hash caiga dentro del rango de índices de la tabla hash.

La función calcularSalario es un ejemplo de cómo se pueden utilizar las tablas de hash para resolver problemas complejos. Esta función toma una descripción de trabajo, la divide en palabras y busca cada palabra en la tabla de hash. Si la palabra se encuentra en la tabla de hash, se suma su valor asociado al salario total. La complejidad de tiempo de esta función es O(N*(L + M)), donde N es el número de palabras en la descripción del trabajo, L es la longitud promedio de las palabras y M es el tamaño de la tabla. Esto se debe a que para cada palabra en la descripción del trabajo, la función calcula el valor hash de la palabra (O(L)) y luego recorre la lista en la posición correspondiente de la tabla de hash (O(M)).

En cuanto a los aprendizajes, este problema reforzó mi comprensión de las tablas de hash y su utilidad en la resolución de problemas del mundo real. También me ayudó a apreciar la importancia de elegir la estructura de datos adecuada para el problema en cuestión. En mi futuro como profesional de la informática, estoy seguro de que las tablas de hash seguirán siendo una herramienta valiosa en mi arsenal. Ya sea que esté trabajando en la optimización de consultas de bases de datos, la implementación de cachés, o la resolución de problemas complejos como este, las tablas de hash son una solución eficiente y efectiva.

Referencias:

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT press.

Knuth, D. E. (1998). The art of computer programming: Sorting and searching (Vol. 3). Pearson Education.

Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley Professional.