ESTRUCTURAS DE DATOS

TRABAJO PRÁCTICO N ° 5

Mapeos, diccionarios y tablas de dispersión

Departamento de Ciencias e Ingeniería de la Computación - U.N.S.
Primer cuatrimestre de 2019

Bibliografía:

[GT] Michael Goodrich & Roberto Tamassia. *Data Structures and Algorithms in Java. Fourth Edition*. John Wiley and Sons. 2006.
[W12] Mark A. Weiss. *Data Structures and Algorithm Anlysis in Java*. Third Edition. Addison-Wesley Pearson Education, Inc. 2012.

Ejercicio 1:

a) Defina el TDA Mapeo como se propone en [GT] documentando apropiadamente las operaciones.
b) Implemente el TDA Mapeo utilizando una lista enlazada. Analice el orden del tiempo de ejecución de las operaciones implementadas. Justifique la elección del tipo de lista elegido.
c) Si el tipo de las entradas del mapeo posee una operación para compararlas por menor, proponga dos implementaciones para el mapeo, una con una lista enlazada y ordenada y otra con un arreglo ordenado en forma ascendente. Analice el orden del tiempo de ejecución de ambas implementaciones.

Ejercicio 2

Se poseen dos correspondencias (mapeos) M1 y M2 cuyas entradas tienen por dominio un número de libreta, y por imagen la nota de una materia (de 0 a 10). Ambos mapeos están implementados con una lista simplemente enlazada y sin celda de encabezamiento.

a) Escriba en Java todas las clases para implementar dichos mapeos.
b) Realice un comando que reciba los dos mapeos M1 y M2 y devuelva una lista L con aquellos elementos E1 de M1 y E2 de M2 que coincidan en el valor del dominio, pero tengan una imagen diferente. Por ejemplo, si E1= (LU: 29303, Nota: 8) pertenece a M1 y E2= (LU:29303, Nota: 7) pertenece a M2, entonces E1 y E2 serán puestos en L.
c) Calcule el orden del tiempo de ejecución del procedimiento implementado. Justifique su respuesta.

Ejercicio 3

✍Suponga que posee una tabla de dispersión abierta (hash externo o, tabla de buckets, o *separate chaining* de acuerdo a [GT][1]) con 6 cubetas (buckets), para almacenar números enteros. Además, posee la función de dispersión h(i) = i mod 6. Muestre la tabla de dispersión resultante de insertar la siguiente secuencia de elementos en una tabla vacía:

(i) 13, 5, 30, 10, 7, 21, 9
(ii) 26, 8, 62, 14, 38, 50

¿Cómo podría redefinir la función de dispersión para evitar el problema que presenta la secuencia (ii) sabiendo que todos los números a insertar serán de la forma N = (6 K) +2?

[1] La tabla de dispersión abierta corresponde a un arreglo de listas de buckets, donde la entrada i-ésima del arreglo contiene una lista con todos los elementos que poseen el mismo valor de "hash" (es decir, los elementos colisionados).

Ejercicio 4

a) Implemente un mapeo de acuerdo a la interfaz Mapeo de [GT] en términos de una tabla de dispersión abierta (en [GT], este tipo de estructura de datos se menciona como arreglo de buckets).
b) Analice el orden del tiempo de ejecución de la implementación dada en el inciso (a).
c) Utilice el mapeo definido en (a) para implementar una función que hace corresponder el número de legajo de un alumno con los datos completos de un alumno. Para ello, suponga que posee una clase Alumno como la definida a continuación:

Alumno
- legajo : String - nombre : String - finales : Lista<ExamenFinal>

Completar constructores, setters y getters habituales, un comando para agregar un final, y un iterador para recorrer los exámenes finales. Discuta implementaciones posibles de la función hashCode para la clase Alumno.

Ejercicio 5

Agregue una operación a la clase programada en el ejercicio (4.a) que convierta a la tabla de dispersión T1 con B cubetas (buckets) en otra tabla de dispersión T2 con B' cubetas tal que B' > 2B y B' es un entero primo, que contenga todos los elementos de la tabla T1.

Ejercicio 6

Implementar el TDA mapeo (correspondencia) utilizando una tabla de dispersión cerrada (en [GT] esta estructura de datos es llamada direccionamiento abierto, u *open addressing*). Utilizar una estrategia de resolución lineal de colisiones.[2]

Ejercicio 7

a) Escriba un método ACOMODAR(D) que recibe un mapeo D, y que para cada elemento del mapeo con imagen I elimine de D todos los demás elementos que tengan imagen I. De esta manera, luego de utilizar el procedimiento, no habrá en el mapeo elementos con imágenes iguales.
b) Justificando su respuesta, indique el orden del tiempo de ejecución del método ACOMODAR asumiendo que:
    - el mapeo está implementado con una lista enlazada.
    - el mapeo está implementado con una tabla de dispersión abierta.
c) Escriba una función CREAR-MAPEO-INVERSO(D) que reciba un mapeo D, y retorne un mapeo inverso M donde estén todas las entradas de D, pero invertidas, esto es, la correspondencia es tal que si en D está el par (A,B), entonces, en M deberá estar el par (B,A). Proponga excepciones para los casos en los que tal mapeo inverso no pueda ser creado.
d) Justificando su respuesta, indique el orden del tiempo de ejecución del método CREAR-MAPEO-INVERSO asumiendo que:
    - el mapeo está implementado con una lista enlazada.
    - el mapeo está implementado con una tabla de dispersión abierta

[2] El hash cerrado corresponde a un arreglo de buckets donde en cada bucket se almacena un único elemento y las colisiones se almacenan en otro bucket de acuerdo a una estrategia en particular (linear probing, quadratic probing, etc.).

Ejercicio 8

Teniendo en cuenta las representaciones de conjuntos de vectores de bits dadas en la teoría, proponga modificaciones adecuadas para resolver los ejercicios de mapeos que se proponen a continuación: escriba en Java la estructura de datos que a su juicio se debería utilizar para optimizar el tiempo de las operaciones *get* y *put* de los siguientes mapeos. Justifique su elección. Indique el espacio en memoria usado y el orden del tiempo de ejecución de las operaciones *get* y *put*. La expresión A $\rightarrow$ B indica que A es el dominio y B el codominio del mapeo.

a) números de libreta de los alumnos en la materia "estructura de datos" $\rightarrow$ sus nombres
b) números de libreta de los alumnos del curso de redes que fue limitado a 200 inscriptos $\rightarrow$ sus nombres
c) códigos de artículos del 20.000 al 20.100 $\rightarrow$ sus precios
d) nombre de mis amigos $\rightarrow$ sus números de teléfono
e) los nombres de los 20 mejores promedios de la UNS $\rightarrow$ sus notas
f) materias de la licenciatura que un alumno tiene aprobadas $\rightarrow$ sus notas
g) días de la semana que salgo a correr $\rightarrow$ la hora
h) días que llovió en Bahía Blanca desde el año 1950 $\rightarrow$ cantidad de mm.
i) días que nevó en Bahía Blanca desde el año 1950 $\rightarrow$ cantidad de cm.

Ejercicio 9

a) Escriba en Java un procedimiento que dados dos mapeos M1 y M2, determine si todas las claves de M1 están contenidas en M2.
b) Indique el orden del tiempo de ejecución del procedimiento implementado. Justifique su respuesta.

Ejercicio 10

¿Qué estructuras de datos o TDAs estructurales pueden utilizarse para implementar un TDA diccionario? Indique ventajas y desventajas de cada una de ellas.

a) Implemente el TDA diccionario utilizando una tabla de hash abierto (*separate chaining*).
b) Implemente el TDA diccionario utilizando una tabla de hash cerrado (*open addressing*).
c) Discuta una implementación de (b) donde se usa resolución lineal de colisiones pero todas las entradas con una misma clave se agrupan en una lista.

Ejercicio 11

a) En un hospital donde hay 100 habitaciones hay un médico clínico encargado para cada habitación. Indique cuál es el TDA que mejor se adapta para mantener información sobre los encargados de cada habitación. Indique cuales son las estructuras de datos posibles para implementar el TDA elegido y escriba en Java la que considere más adecuada para este caso particular.
b) Indique cuál es el TDA que mejor se adapta para mantener información sobre los enfermeros de un hospital que se dividen en tres turnos (Mañana, Tarde y Noche) y cada enfermero tiene asociado un grupo de habitaciones. Indique cuales son las estructuras de datos posibles para implementar el TDA elegido y escriba en Java la que considere más adecuada para este caso particular.
c) Indique cuál es el TDA que mejor se adapta para mantener información sobre los pacientes que ingresan a la Guardia. ¿Si se los quisiera organizar según el médico que los debería atender qué TDAs utilizaría?
d) Asumiendo que los TDA de los incisos anteriores fueron implementados, y utilizando las operaciones vistas en clase para los TDA elegidos. Escriba en Java una operación que dado un enfermero devuelva que médicos encargados le asignan tareas para las habitaciones.

e) Suponiendo que dispone del TDA lista y asumiendo que los TDA de los incisos anteriores fueron implementados. Utilizando las operaciones vistas en clase para los TDA elegidos, escriba en Java una operación que dada una lista de enfermeros devuelva todas las habitaciones en que trabajan.
f) Se descubrió que un paciente tiene un virus peligroso y se desea rastrear un posible contagio. Escriba una operación que, dada la habitación de un paciente, devuelva una lista de enfermeros que lo atendieron.
g) Indique el orden del tiempo de ejecución de los procedimientos implementados. Justifique.

Ejercicio 12

Compare las operaciones del TDA diccionario implementado con:

a) una tabla de dispersión
b) una lista enlazada no ordenada
c) una lista enlazada ordenada
d) un arreglo

Analice el orden del tiempo de ejecución de las operaciones y el espacio utilizado en memoria.

Ejercicio 13

El tipo conjunto de elementos de tipo genérico E tiene las siguientes operaciones:

- insertar un elemento de tipo E,
- eliminar un elemento de tipo E,
- determinar si un elemento de tipo E pertenece al conjunto,
- realizar la unión con otro conjunto B actualizando el conjunto receptor del mensaje,
- realizar la intersección con otro conjunto B actualizando el conjunto receptor del mensaje.

Plantee una interfaz acorde a la especificación dada y determine cómo implementar un conjunto de elementos de tipo E usando:

a) una tabla de dispersión
b) una lista enlazada no ordenada
c) una lista enlazada ordenada
d) un arreglo

Analice el orden del tiempo de ejecución de las operaciones y el espacio utilizado en memoria.

Ejercicio 14

Un Bag es una colección de elementos que soporta elementos repetidos. Dada la siguiente especificación para el TDA Bag que extiende la interfaz Iterable<E>:

<<interface>> Bag<E>
//inserta un ítem en el Bag void add(E item); *//Retorna verdadadero si el bag está vacío y falso en caso contrario.* boolean isEmpty(); *//retorna la cantidad de elementos almacenados en el bag* int size(); *//elimina una aparición del elemento especificado del bag* void remove(E item); *//retorna la cantidad de apariciones del elemento en el bag.* int getCount(E ítem);

Implementar la interfaz Bag usando dos representaciones posibles:

a) una lista doblemente enlazada.
b) un diccionario.