

TAD Tablas de Dispersión

Práctica 17

Tabla de dispersión con direccionamiento abierto

Objetivo

El objetivo de esta práctica es que el alumno refuerce sus conocimientos acerca de tablas de dispersión mediante el desarrollo de algunos métodos para una implementación del TAD tablas de dispersión.

Descripción general

La práctica consiste en completar una implementación del TAD tablas de dispersión que usa la técnica de direccionamiento abierto y prueba lineal para resolver colisiones. La particularidad de esta implementación es que deberá soportar la función de eliminación de elementos.

La técnica para **eliminar** elementos será la de marcar con la etiqueta "BORRADO" el lugar en el arreglo que corresponda al elemento eliminado. De esta forma el método para agregar elementos buscará por un lugar vacío o marcado con la etiqueta para poder insertar el nuevo elemento. Mientras que los métodos de búsqueda, **contiene** y **obtener**, no deberán parar su búsqueda cuando se topen con una lugar marcado con dicha etiqueta.

Se realizará un reajuste de la capacidad de la tabla y por ende la reasignación de sus elementos. En el método **agregar** se duplicará la capacidad de la tabla cuando el factor de carga sea mayor o igual a 0.5, es decir, cuando su porcentaje de ocupación sea al menos el 50%. Mientras que en el método **eliminar** la capacidad se reducirá a la mitad cuando el factor de carga sea menor o igual a 0.125, es decir, cuando sólo un 12.5% de la tabla se encuentre ocupada. De ésta forma se evitará que los métodos de búsqueda se vuelvan lentos. Recordar que el factor de carga es la cantidad de elementos en la tabla entre el tamaño de ella.

Material

El material de esta práctica consta de los siguientes archivos:

- **InterfazTablaDispersion.class** interfaz para trabajar con tablas de dispersión.
- **TablaDispersionA2.java** clase que implementará el TAD tablas de dispersión con direccionamiento abierto y prueba lineal. En esta clase se incluirán los métodos solicitados.

- **PruebaTablaDispersion.class** programa para probar la clase `TablaDispersionA2.java`.
- Documentación:
 - `InterfazTablaDispersion.html` documentación de la clase `InterfazTablaDispersion`.

Desarrollo

1. Descargar los archivos `InterfazTablaDispersion.class`, `TablaDispersionA2.java` y `PruebaTablaDispersion.class` en el directorio donde se va a desarrollar la práctica.
2. Leer la documentación y revisar los archivos del material de la práctica. Recordar que la firma de los métodos proporcionados y requeridos en el desarrollo de la práctica no podrán ser modificados.
3. Dentro de la clase `TablaDispersionA2` se implementarán los siguientes métodos:
 - a. **agregar(Object o)** método para incluir un nuevo elemento. Antes de agregar el nuevo elemento, duplica la capacidad de la tabla y reasigna sus elementos si el factor de carga es mayor o igual a 0.5.
 - b. **borrar(Object o)** método para eliminar de la tabla de dispersión el elemento indicado como parámetro. Una vez que se elimina el elemento, reduce a la mitad la capacidad de la tabla y reasigna sus elementos si el factor de carga es menor o igual a 0.125. Si el elemento no se encuentra en la tabla, no hace nada.
 - c. **contiene(Object o)** devuelve true si el elemento buscado se encuentra en la tabla. En otro caso devuelve false.
 - d. **obtener(Object o)** devuelve el elemento buscado si se encuentra en la tabla. En otro caso devuelve null.
 - e. **iterador()** método que devuelve un iterador para recorrer los elementos de la tabla de dispersión. El iterador no debe soportar la operación *remove*.
4. Probar la implementación realizada ejecutando el programa `PruebaTablaDispersion`. Si el programa funciona adecuadamente se verán los siguientes mensajes:

Se crea tabla de dispersion con capacidad indicada de 15 elementos

Capacidad tabla: 17

```
¿Esta vacia?: true
Num de elementos: 0
```

Se agregan las siguientes letras: Y, E, A, R, J, L, S, C

```
Num de elementos: 8
Iterando sobre la tabla de dispersion:
```

```
C
E
Y
J
L
A
R
S
```

```
---¿Contiene la letra A?: true
Obtener la letra A: A
Eliminando la letra A...
Num de elementos: 7
```

```
¿Contiene la letra A?: false
Obtener letra A: null
Tratar de eliminar letra inexistente, letra A...
Num de elementos: 7
Imprime contenido de la tabla:
0:C 1:E 2:null 3:null 4:Y 5:null 6:J 7:null 8:L 9:null
10:null 11:null 12:null 13:null 14:BORRADO 15:R 16:S
```

```
---Prueba de reajuste de la capacidad de la tabla al eliminar
elementos.
Eliminando las siguientes letras S, C, E, A, Y y L...
Nueva capacidad de la tabla: 11
Num de elementos: 2
Imprime contenido de la tabla:
0:null 1:null 2:null 3:null 4:null 5:R 6:null 7:null 8:J
9:null 10:null
```

```
---Prueba de reajuste de la capacidad de la tabla al agregar
elementos.
Agregar las letras L, G, W, Y
```

```
Nueva capacidad de la tabla: 23
Num de elementos: 6
Imprime contenido de la tabla:
0:null 1:null 2:G 3:null 4:null 5:J 6:null 7:L 8:null
9:null 10:null 11:null 12:null 13:R 14:null 15:null 16:null
17:null 18:W 19:null 20:Y 21:null 22:null
```

Ejercicio opcional

Se propone desarrollar una variación de esta estructura de datos en la que se utilice la técnica de direccionamiento abierto con un manejo de colisiones que haga uso de una prueba cuadrática.