



Estructura de Datos y Algoritmos

ITBA 2024-Q1



Vamos a usar un arreglo ordenado como soporte para un índice.

TP 3A- Ejer 1

Crear el índice con los siguientes servicios y chequear con Junit

Leer bien la especificación sobre el insert

```
package eda;
```

```
public interface IndexService {
```

```
// elements serán los valores del índice, los anteriores se descartan.  
// lanza exception si elements is null y deja los valores anteriores.  
void initialize(int [] elements);
```

```
// busca una key en el índice, O(log2 N)  
boolean search(int key);
```

```
// inserta el key en pos correcta. Crece automáticamente de a chunks  
void insert(int key);
```

```
// borra el key si lo hay, sino lo ignora.  
// decrece automáticamente de a chunks  
void delete(int key);
```

```
// devuelve la cantidad de apariciones de la clave especificada  
int occurrences(int key);
```

```
}
```



Caso de Uso:

```
IndexService myIndex= new IndexWithDuplicates();
```

```
System.out.println (myIndex.occurrences( 10 ) ); // se obtiene 0
```

```
myIndex.delete( 10 ); // ignora
```

```
System.out.println (myIndex.search( 10 ) ); // se obtiene false
```

```
myIndex.insert( 80 ); // almacena [80]
```

```
myIndex.insert( 20 ); // almacena [20, 80]
```

```
myIndex.insert( 80 ); // almacena [20, 80, 80]
```

```
try
{
    myIndex.initialize( new int[] {100, 50, 30, 50, 80, 100, 100, 30} );
}
catch(Exception e)
{
}
```

// el índice posee [30, 30, 50, 50, 80, 100, 100, 100]

`System.out.println(myIndex.search(20));` *// se obtiene **false***

`System.out.println(myIndex.search(80));` *// se obtiene **true***

`System.out.println (myIndex.occurrences(50));` *// se obtiene **2***

`myIndex.delete(50);`

`System.out.println (myIndex.occurrences(50));` *// se obtiene **1***

...

Detalles de implementación:

- declarar una constante con el tamaño de **chunksize** por el que irá creciendo el arreglo.
- para crecer el arreglo usar **Arrays.copyOf**
- implementar un método privado **getClosestPosition** que devuelve:
 - si no existe la key la posición donde insertarla
 - si existe la key la posición de alguna de las repeticiones

Quiero insertar el 25

Índice al inicio

2	8	10	15	17	21	28	30	34	42	50			
---	---	----	----	----	----	----	----	----	----	----	--	--	--

GetClosestPosition



2	8	10	15	17	21	28	30	34	42	50			
---	---	----	----	----	----	----	----	----	----	----	--	--	--

Muevo los elementos



2	8	10	15	17	21	28	30	34	42	50	50		
---	---	----	----	----	----	----	----	----	----	----	----	--	--

Resultado después de mover los elementos

2	8	10	15	17	21	28	28	30	34	42	50		
---	---	----	----	----	----	----	----	----	----	----	----	--	--

Inserto en la posición correcta



2	8	10	15	17	21	25	28	30	34	42	50		
---	---	----	----	----	----	----	----	----	----	----	----	--	--



¿Qué complejidad temporal y espacial tienen los algoritmos propuestos?