

Árboles de Búsqueda Binarios

Ejemplos de aplicación

Ejemplo 1

- El dataset **Goodread-books** contiene una amplia lista de libros con sus datos principales.
- Nos interesa poder consultar información como por ejemplo:
 - Cuántos libros tiene cada autor?
 - Qué calificación promedio tienen los libros de un autor?
 - Se quiere obtener una estadística del número de libros por año
 - Obtener el libro mejor calificado de cada autor

Solución 1: ADT Book

```
class Book {  
    private int bookID;  
    private String title;  
    private String authors;  
    private float average_rating;  
    private String isbn;  
    private String isbn13;  
    private String language_code;  
    private int num_pages;  
    private int ratings_count;  
    private int text_reviews_count;  
    private Date publication_date;  
    private String publisher;  
  
    Book(String line) throws ParseException { ... }  
}
```

Solución 2: Leer los datos

```
public static void readFile(String file) {  
    In in = new In(file);  
    in.readLine();           // ignorar primera linea  
    while(! in.isEmpty()) {  
        String l = in.readLine();  
        try {  
            Book b = new Book(l);  
            ...  
        }  
        catch(ParseException e) {  
            StdOut.println("Fecha no valida: "+l);  
        }  
        catch(NumberFormatException e) {  
            StdOut.println("Numero no valido: "+l);  
        }  
    }  
}
```

Solución 3: Representación en memoria

- Criterio de búsqueda: autor
- Usar un BST con llave el autor
- Cuál sería el valor?

BST<String, ?>

- Y cómo se sacaría la estadística por fechas?

BST<?, ?>

Solución 4: Estructuras a utilizar

```
BST<String, Bag<Book>> librosPorAutor = new BST<>();
```

```
BST<Integer, Integer> cantidadPorAño = new BST<>();
```

Solución 5: Carga de datos

```
Book b = new Book(1);

// Integer año = b.getPublication_date().getYear();
GregorianCalendar gc = new GregorianCalendar();
gc.setTime(b.getPublication_date());
Integer año = gc.get(Calendar.YEAR);

if (! librosPorAutor.contains(b.getAuthors()) )
    librosPorAutor.put(b.getAuthors(), new Bag<Book>());

if (! cantidadPorAño.contains(año))
    cantidadPorAño.put(año, 0);

librosPorAutor.get(b.getAuthors()).add(b);
cantidadPorAño.put(año, cantidadPorAño.get(año)+1);
```

Solución 6: Consultas por autor

```
public static void librosPorAutor(String autor) {  
    Bag<Book> libros = librosPorAutor.get(autor);  
    double[] ratings = new double[libros.size()];  
    int i=0;  
    for(Book b: libros) {  
        StdOut.println(b.getTitle());  
        ratings[i] = b.getAverage_rating();  
        i++;  
    }  
    StdOut.println("Numero de libros: "+libros.size());  
    StdOut.println("Calificacion promedio: "+StdStats.mean(ratings));  
}
```


Solución 7: Estadística por año

```
public static void librosPorAnno() {  
    for(Integer año: cantidadPorAño.keys()) {  
        StdOut.println(año+" : "+cantidadPorAño.get(año));  
    }  
}
```

Mejor libro por autor

Puede haber varios caminos de solución

1. Recorrer la bolsa de libros por autor para obtener el mayor.
2. Utilizar una cola de prioridad MaxPQ para obtener el mayor.
3. Almacenar un árbol aparte donde cada autor tenga una cola de prioridad

Solución 8: Mejor libro por autor

```
private static final BST<String, MaxPQ<Book>> topPorAutor = new BST<>();
```

```
public static void readFile(String file) {  
    ...  
    if (! topPorAutor.contains(b.getAuthors()))  
        topPorAutor.put(b.getAuthors(),  
            new MaxPQ<Book>(new BookComparator()));  
    topPorAutor.get(b.getAuthors()).insert(b);  
    ...  
}  
  
public static void topPorAutor() {  
    for(String autor: topPorAutor.keys()) {  
        Book top = topPorAutor.get(autor).max();  
        StdOut.println(autor+": "+top.getTitle()+" "  
            +top.getAverage_rating());  
    }  
}
```

Ejercicios

- Obtener los dos mejores libros de cada autor.
- En que años se tuvieron las puntuaciones promedio más altas?
- Cuántos libros hay por cada idioma?
- Obtener los 10 mejores libros de toda la colección.
- Qué editor(publisher) tiene el promedio más alto de libros publicados?