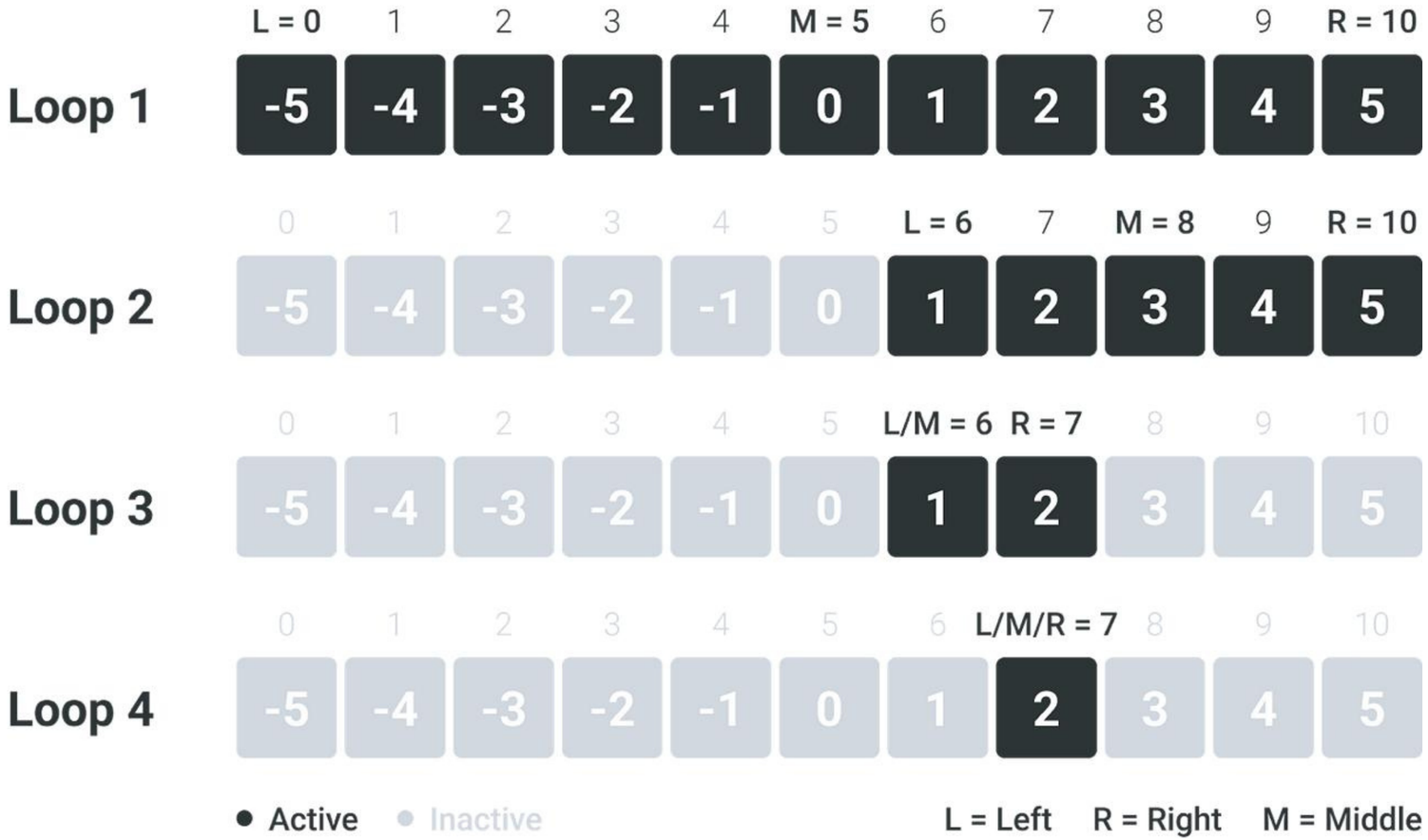
Búsqueda Binaria y métodos de ordenamiento

PROFESOR: YERKO ORTIZ
AYUDANTE DE LAB: DIEGO BANDA



Búsqueda Binaria

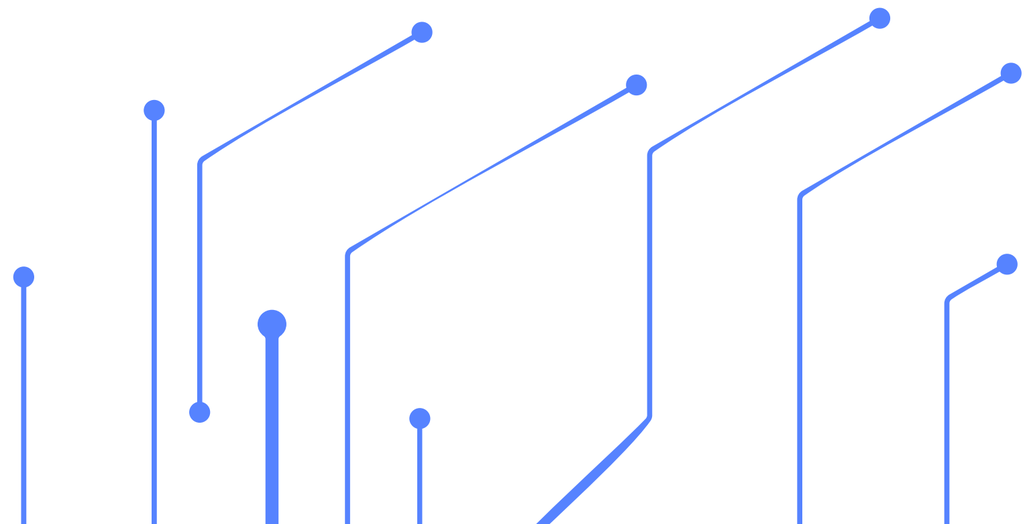
Ejercicio

DISEÑE UN ALGORITMO DE BÚSQUEDA BINARIA, EL CUAL, SI ENCUENTRA EL NÚMERO BUSCADO, RETORNARÁ “TRUE”, EN CASO CONTRARIO, RETORNARÁ “FALSE”.

EL CÓDIGO DEBERÁ DE CONTAR CON DOS FUNCIONES:

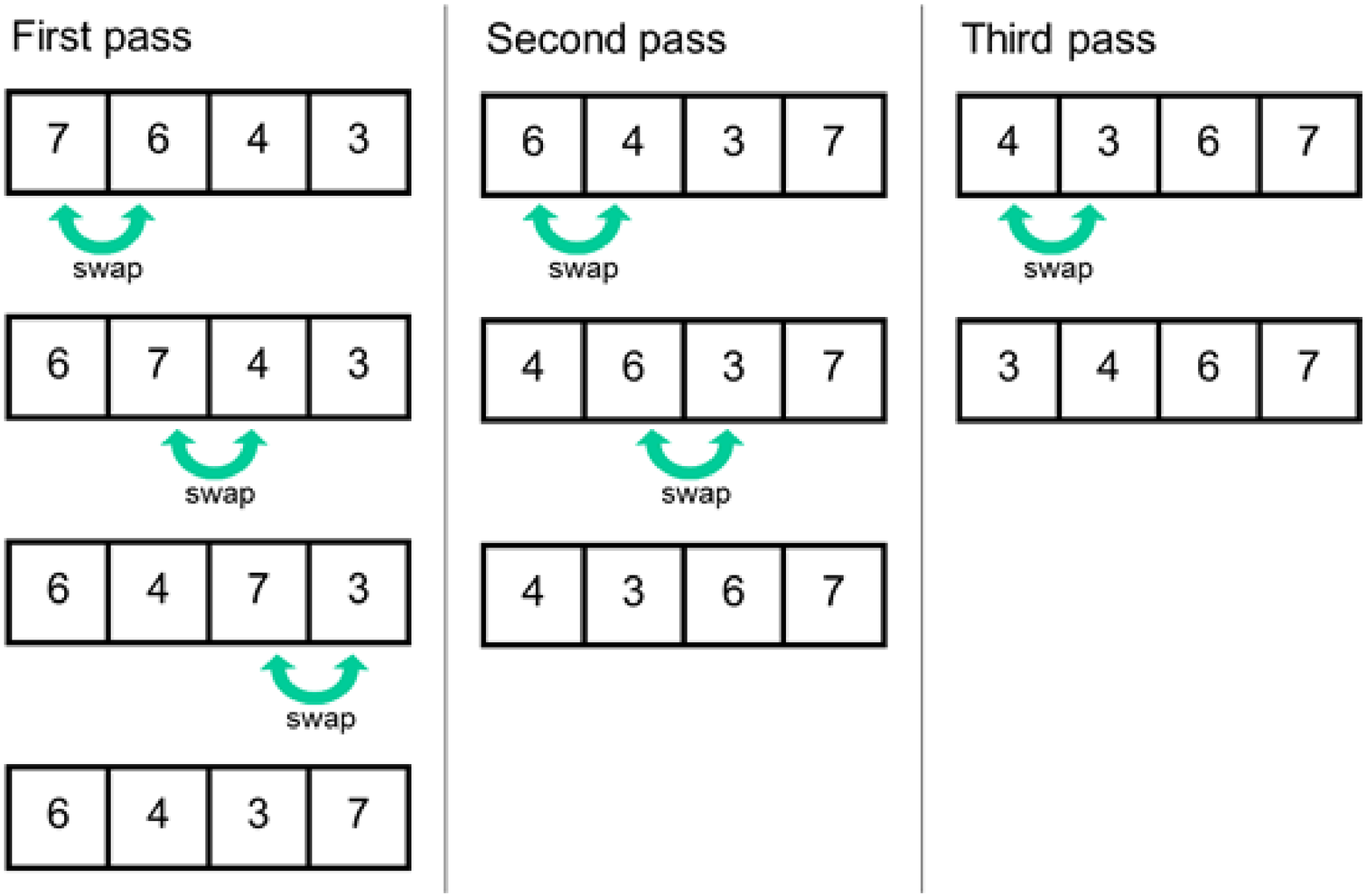
- **Búsqueda binaria iterativa.**
- **Búsqueda binaria recursiva.**

Sorting



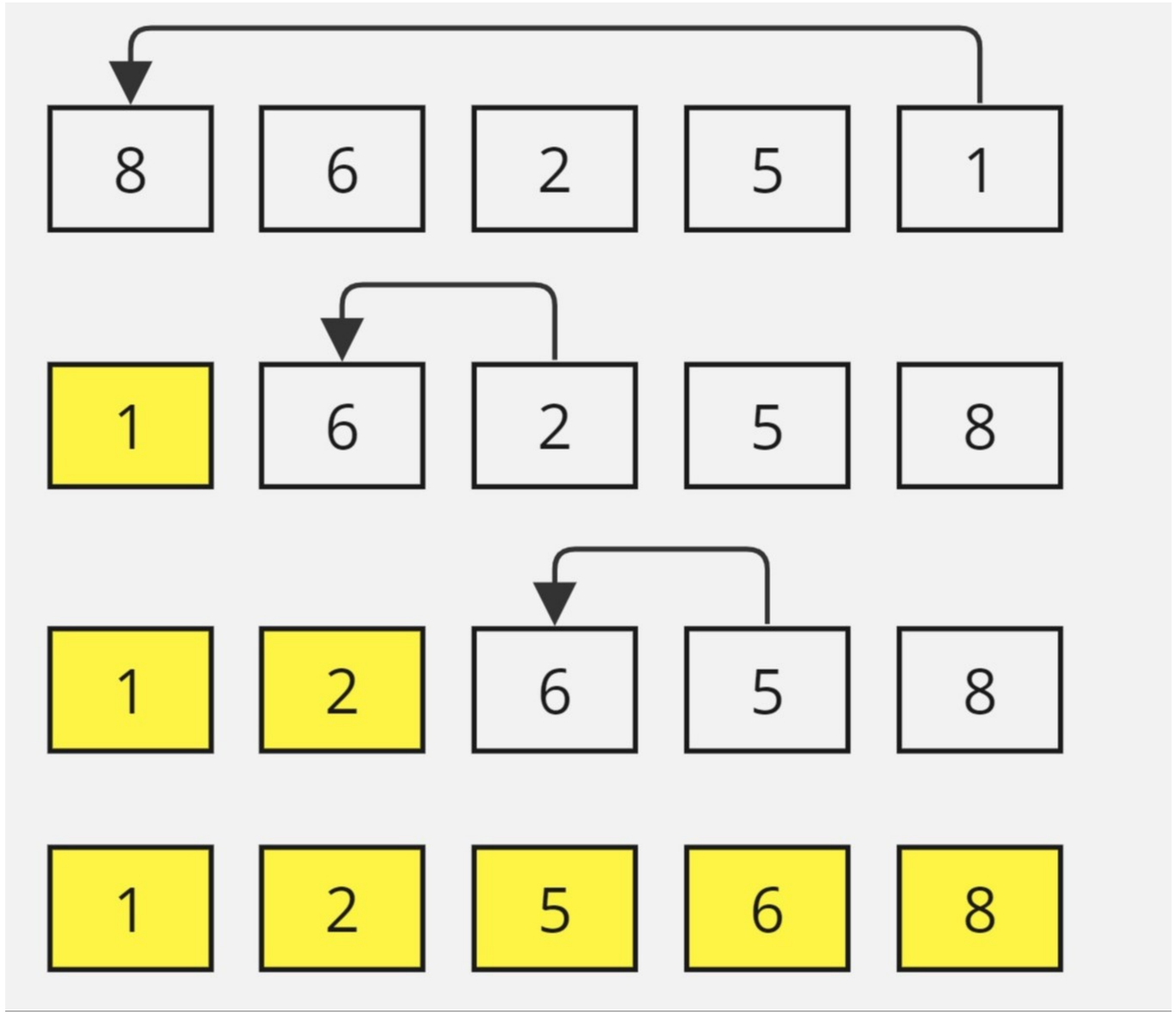
- Bubble sort
- Selection sort
- Insertion sort

Bubble sort


First pass
7 6 4 3
swap
6 7 4 3
swap
6 4 7 3
swap
6 4 3 7
Second pass
6 4 3 7
swap
4 6 3 7
swap
4 3 6 7
Third pass
4 3 6 7
swap
3 4 6 7

bubbleSort.java > ...

```java
public class bubbleSort {
    public static int[] sort(int arr[]){
        for(int i = 0 ; i < arr.length - 1 ; i++){
            for(int j = 0 ; j < arr.length - 1 ; j++){
                if(arr[j] > arr[j + 1]){
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        return arr;
    }
```

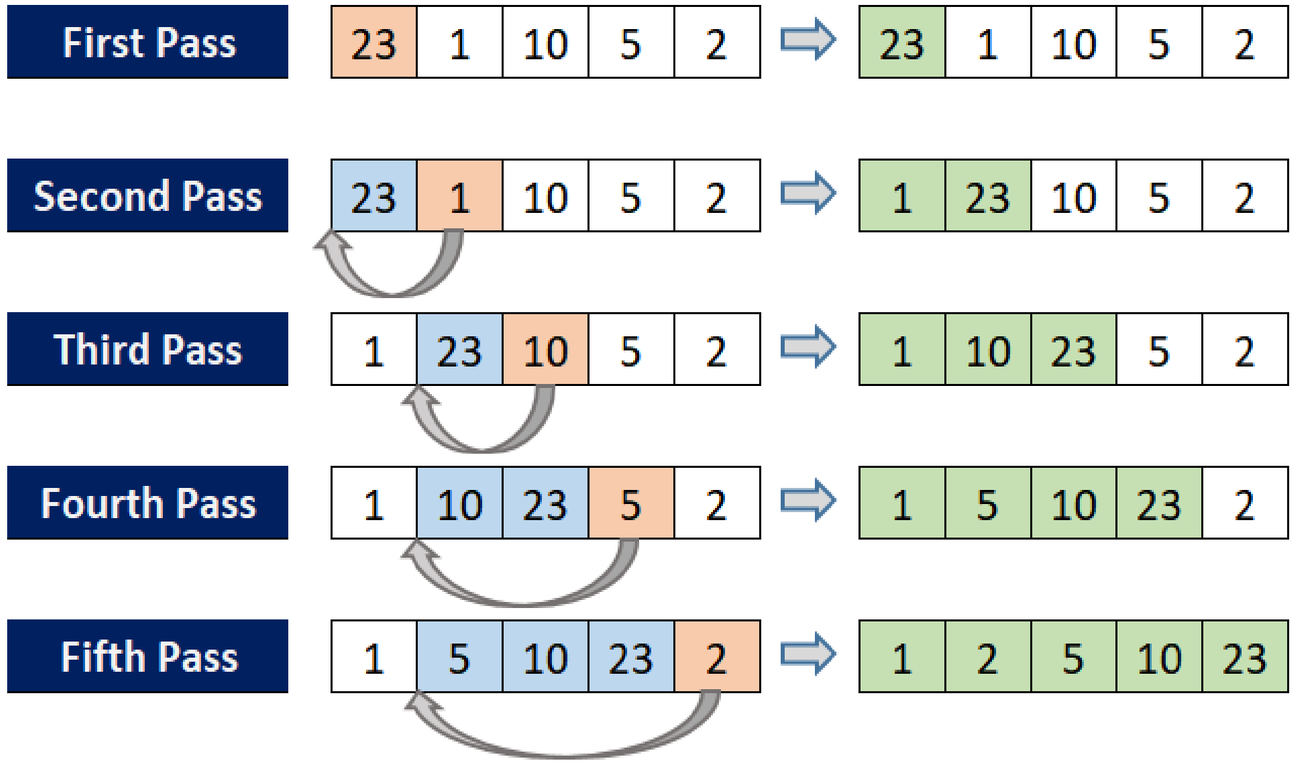


Selection sort

selectionSort.java > ...

```java
public class selectionSort {
    public static int[] sort(int arr[]){
        for(int i = 0 ; i < arr.length - 1 ; i++){
            int min = i;
            for(int j = i + 1; j < arr.length ; j++){
                if(arr[min] > arr[j]){
                    min = j;
                }
            }
            int temp = arr[min];
            arr[min] = arr[i];
            arr[i] = temp;
        }
        return arr;
    }
}
```

Insertion sort



J insertionSort.java > ...

```java
public class insertionSort {
    public static int[] sort(int[] arr){
        for(int i = 1 ; i < arr.length ; i++){
            int key = arr[i];
            int j = i - 1;
            while(j >= 0 && arr[j] > key){
                arr[j+1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
        return arr;
    }
}
```