

# Taller de Segundo Corte: Recursión, Ordenamiento y Búsquedas

Prof. Luis Garreta

Curso de Estructuras de Datos y Algoritmos I  
Universidad Autonoma de Occidente – Cali  
Depto. Operaciones y Sistemas  
Facultad de Ingeniería

24 de marzo de 2018

1. Muestre paso a paso como se realiza el ordenamiento del arreglo A: [6,3,2,7,4,1,5] por los algoritmos de MergeSort, QuickSort, BubbleSort, SelectionSort, e InsertionSort: A: [6,3,2,7,4,1,5]
2. Lo mismo que el punto anterior pero ahora para la cadena "colombia". Asuma un orden lexicográfico (a..z) y que la cadena empieza en la posición 0, es decir suponga el arreglo de caracteres A: ['c','o','l','o','m','b','i','a']
3. Realice el seguimiento de los llamados recursivos del algoritmo de ordenamiento MergeSort cuando ordena el arreglo A: [5,2,1,4,3]
4. Implemente una función recursiva que dado como entrada una cadena de caracteres determine si está es palíndromo o no. Una cadena palíndromo es aquella que se puede leer igual de izquierda a derecha como de derecha a izquierda. Por ejemplo: "OSO", "AEREA", "AVIVA", "SELLES", "SARRAS". El prototipo de la función es el siguiente:

```
static boolean esPalindrome (String palabra, int ini, int fin)
```

donde ini y fin son las posiciones inicial y final desde donde se van a comparar los arreglos, que al principio serían 0 y n-1, si los arreglos tienen n elementos
5. Implemente una función recursiva que retorne true/false si dos arreglos de la misma longitud son iguales. El prototipo de la función es el siguiente:

```
static boolean igualesArreglos (int [] A1, int [] A2, int ini, int fin)
```

donde ini y fin son las posiciones inicial y final desde donde se van a comparar los arreglos, que al principio serían 0 y n-1, si los arreglos tienen n elementos
6. Los algoritmos de búsqueda más usados son la binaria y la secuencial. La binaria es más eficiente cuando la cantidad de datos es muy grande, mientras que la secuencial es más eficiente cuando son pocos datos. Teniendo en cuenta lo anterior, es decir, cuando los datos sean menores que un valor M, la búsqueda se hace de forma secuencial, de lo contrario se hace binaria, diseñe un nuevo algoritmo de búsqueda recursivo que combine las dos ideas (secuencial y binaria). Al algoritmo le ingresa un arreglo A, las posiciones INI y FIN donde inicia y termina la búsqueda, el elemento ELEM a buscar, y el valor M para realizar la búsqueda secuencial.  
Por ejemplo: suponga que al algoritmo le ingresa un arreglo de cadenas A={"casa", "jarro",..., "xerox", "zorro"}, INI=0, FIN=100 millones, y M=10000. Este debería iniciar partiendo el arreglo de 100 millones hasta que en algún momento se forme un subarreglo de 10000 elementos, en ese preciso momento se debería realizar la búsqueda secuencial sobre los 10000 elementos y retornar la respuesta de si está el elemento o no en ese subarreglo. Si el elemento está, entonces termina la búsqueda y retorna la posición donde lo encontró, y si no está debería continuar la búsqueda de acuerdo a la lógica de la búsqueda binaria. Al final, si no lo encuentra en ningún lado debería retornar -1. El prototipo de la función es:

```
static int busquedaIterRecursiva (int [] A, int INI, int FIN, int ELEM, int M)
```
7. Realice el seguimiento (como se están llevando los llamados recursivos) para el llamado del algoritmo anterior con los siguientes valores: A={1,2,3,4,5,6,7,8,9,10,11,12}, INI=0, FIN=7, ELEM=5, y M=3.