

Ordenamiento Recursivo MergeSort (Mezclas)

Curso de Estructuras de Datos

Prof. Luis E. Garreta U.

lgarreta@uao.edu.co

Universidad Autonoma de Occidente – Cali

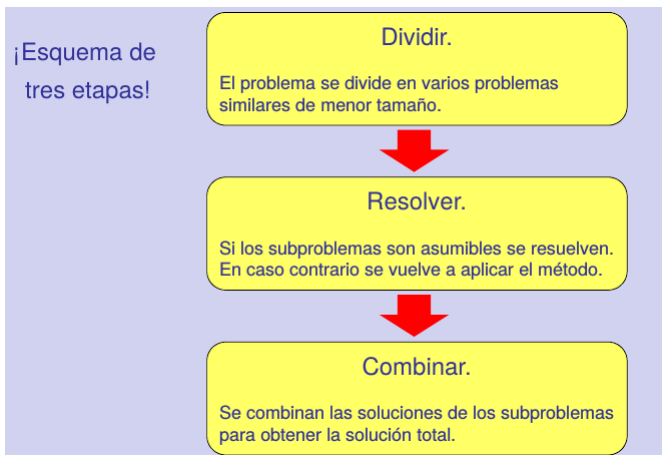
Depto. Operaciones y Sistemas

Facultad de Ingeniería

6 de marzo de 2018

Divide y Vencerás

Una de las estrategias más conocidas para el diseño de algoritmos:



Esquema Divide y Vencerás



Características

Características que deben cumplir los problemas para que se pueda aplicar esta técnica:

- ▶ El problema se debe poder descomponer en otros similares pero más pequeños.
- ▶ Los nuevos problemas deben ser disjuntos.
- ▶ Debe ser posible combinar las soluciones individuales para obtener la global.

Ejemplo 1: *maxArreglo* iterativo

Encontrar el valor máximo en un arreglo de enteros

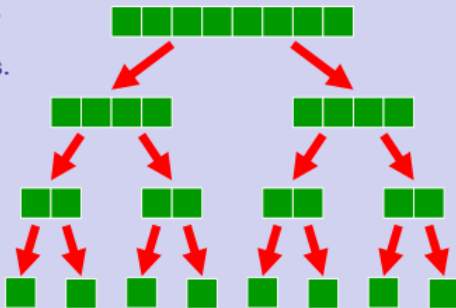
A : [11, 21, 22, ..., .. 9, 8, 55]

```
int maxArreglo (int A[], int n) {  
    int maximo = A [0];  
    for (int i=1; i<n; i++) {  
        if (A[i] > maximo)  
            maximo = A[i]  
  
    return maximo;  
}
```

Enfoque Divide y Vencerás *maxArreglo*

1. Dividir el array en dos partes.
2. Hallar el máximo de cada parte.
3. Seleccionar el mayor de los dos.

Se sigue aplicando
de forma recursiva.



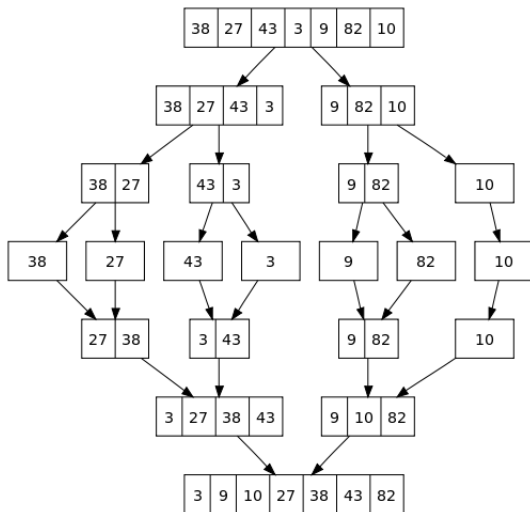
Ejemplo 1: *maxArreglo divide y vencerás*

```
int maxArreglo (int A[], int ini, int fin) {
    int mitad, maxIzquierda, maxDerecha;
    if (ini == fin)
        return A [ini];
    else {
        mitad = (ini+fin)/2;
        maxIzquierda = maxArreglo (A, ini, mitad);
        maxDerecha   = maxArreglo (A, mitad+1, fin);
        if (maxIzquierda > maxDerecha)
            return maxIzquierda;
        else
            return maxDerecha;
    }
    return -1;    // Buena prActica por si no retorna nada
}
```

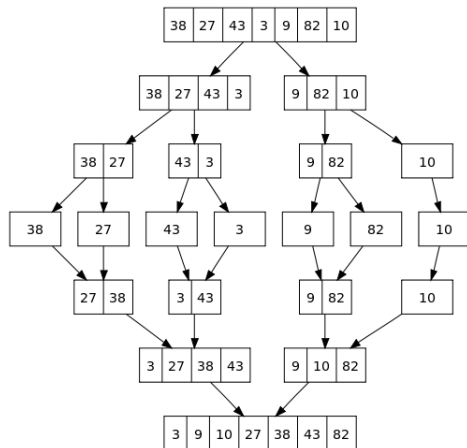
Ejemplo 2: Búsqueda Binaria Divide y Vencerás

```
int busquedaBinaria (int A[], int ini, int fin, int k) {  
    int mitad;  
    if (fin >= ini) {  
        mitad = ini + (fin-ini)/2;  
        if (k == A [mitad])  
            return mitad;  
        if (k < A [mitad])  
            return busquedaBinaria (A, ini, mitad-1, k);  
        else  
            return busquedaBinaria (A, mitad+1, fin, k);  
    }  
    return -1;  
}
```


Ejemplo 2: Ordenamiento Merge Sort



Mergesort Dividir y Vencerás



DIVIDIR: Dividir el arreglo a ordenar de n elementos en dos arreglos de tamaño $n/2$

CONQUISTAR: Ordenar recursivamente los dos subarreglos usando Mergesort

COMBINAR: Mezclar los dos subarreglos ordenados para producir un nuevo arreglo ordenado

Función Merge de Mergesort

- **Input:** Sorted arrays $K[1..n_1]$, $L[1..n_2]$
- **Output:** Merged sorted array $M[1.. n_1+n_2]$

```
i = 1, j = 1
for t = 1 to  $n_1 + n_2$ 
  { if ( $i \leq n_1$  and ( $j > n_2$  or  $K[i] < L[j]$ ) )
    then {  $M[t] = K[i]$ ,  $i = i + 1$  }
    else {  $M[t] = L[j]$ ,  $j = j + 1$  }
  }
```

Linear Time Complexity: $\Theta(n_1 + n_2)$

Algoritmo MergeSort

Merge-Sort $A[1 \dots n]$

If $n > 1$ then

1. Recursively merge-sort $A[1 \dots \lfloor n/2 \rfloor]$
and $A[\lfloor n/2 \rfloor + 1 \dots n]$
2. Merge the two sorted subsequences