

MergeSort, parte 1

MergeSort

El funcionamiento es el siguiente:

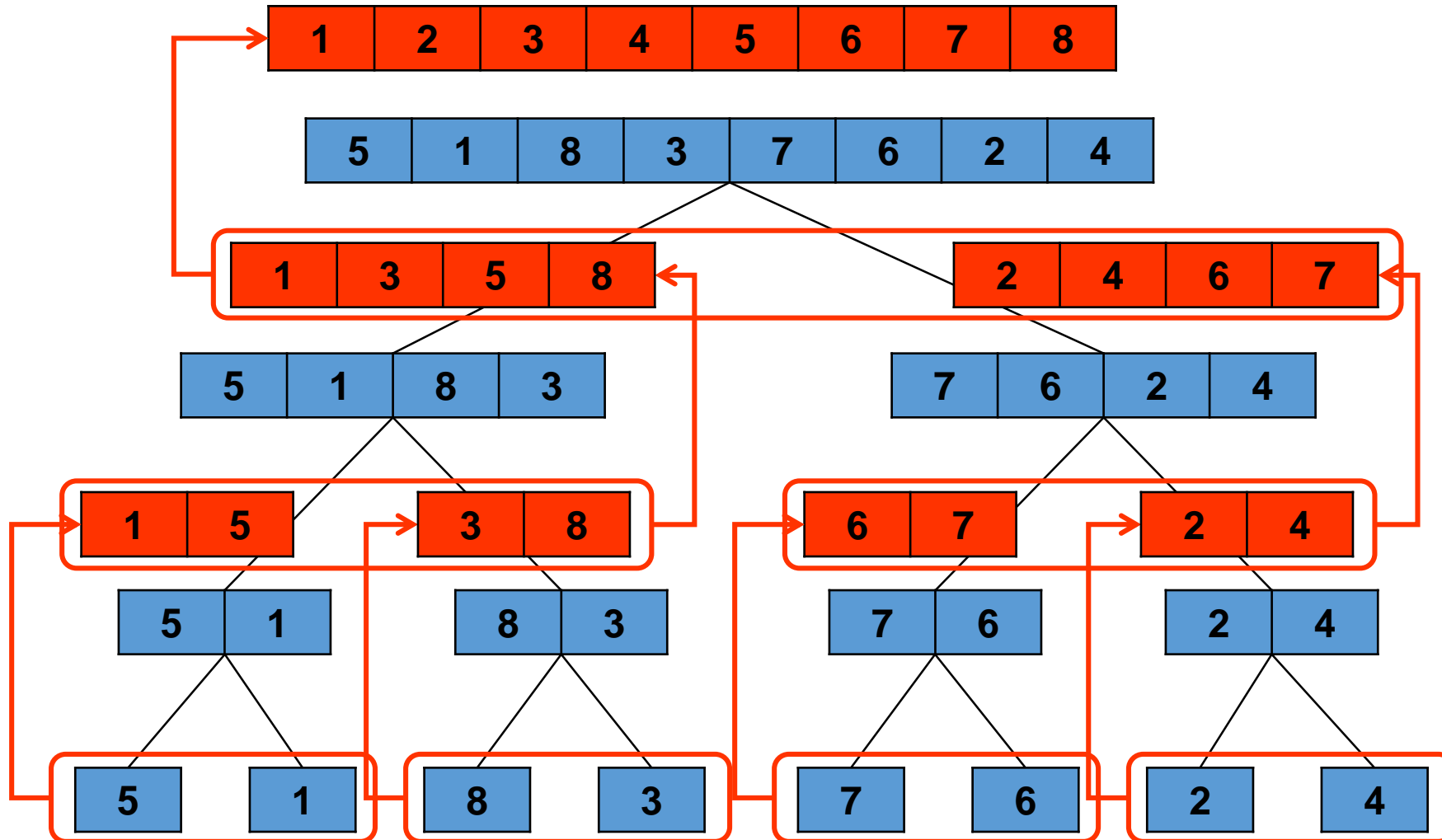
- Si la longitud del arreglo es 0 ó 1, entonces ya está ordenado. En otro caso:
- Separar el arreglo en dos sub-arreglos de aproximadamente la mitad del tamaño.
- Ordenar cada sub-arreglo recursivamente.
- Mezclar los dos sub-arreglos ya ordenados en un solo arreglo ordenado y retornarlo.

Este algoritmo se basa en dos ideas:

- Un arreglo pequeño necesitará menos pasos para ordenarse que uno grande.
- Se necesitan menos pasos para construir un arreglo ordenado a partir de dos subarreglos ordenados, que a partir de dos desordenados.

MergeSort

Veamos el siguiente ejemplo:



MergeSort

```
function mergeSort(X, M):  
    if M > 1:  
        XL = mitad izquierda de X  
        XR = mitad derecha de X  
        return merge(mergeSort(XL, M/2), mergeSort(XR, M/2), M)  
    else  
        return X
```

MergeSort

```
function merge(XL, XR, M){  
  i,j,k = 0  
  while k < M:  
    if  $XL_i \leq XR_j$   
       $Z_k = XL_i$   
       $i++$   
    else if  $XL_i > XR_j$   
       $Z_k = XR_j$   
       $j++$   
    else if  $i \geq M/2$   
       $Z_k = XR_j$   
       $j++$   
    else  
       $Z_k = XL_i$   
       $i++$   
     $k++$ 
```

