

Merge Sort

The *Merge* sort works by recursively sorting the two halves of the given array into some auxiliary data structure (e.g. two other arrays, each half the size of the original array) and then merging the sorted halves back into the original array. As with the Quick sort, the recursion ends when there is just one value to be sorted. For example, assume that the array initially contains the following eight values:

		Recursively split up into groups											
		4	6	7	0	3	9	3	6	Sptart with1 group of eight			
		4	6	7	0			3	9	3	6	Split into 2 groups of 4	
		4	6		7	0		3	9		3	6	Split into 4 groups of 2
4		6	7	0		3		9	3		6		Split into 8 groups of 1
		Recursion ends - Merge groups back together											
		4	6		0	7		3	9		3	6	Merged into 4 groups of 2
		0	4	6	7			3	3	6	9		Merged into 2 groups of 4
		0	3	3	4	6	6	7	9				Merged into 1 group of 8

The *Merge* sort requires $O(n \log n)$ time in all cases, and is more efficient than the Selection sort and usually more efficient than the Insertion sort (except when the array is already sorted or is close to being sorted). However, a disadvantage of the *Merge* sort is that it requires more space than other sorting algorithms because of the auxiliary data structures that it uses.