Implementation of Merge Sort

Merge sort is a recursive algorithm that continually splits a list in half. If the list is empty or has one item, it is sorted by definition (the base case). If the list has more than one item, we split the list and recursively invoke a merge sort on both halves. Once the two halves are sorted, the fundamental operation, called a merge, is performed. Merging is the process of taking two smaller sorted lists and combining them together into a single, sorted, new list.

Resources for Review

Check out the resources below for a review of Merge sort!

- Wikipedia
- Visual Algo
- Sorting Algorithms Animcation with Pseudocode

In [3]:
```
def merge_sort(arr):

    if len(arr)>1:
        mid = len(arr)//2
        lefthalf = arr[:mid]
        righthalf = arr[mid:]

        merge_sort(lefthalf)
        merge_sort(righthalf)

        i=0
        j=0
        k=0
        while i < len(lefthalf) and j < len(righthalf):
            if lefthalf[i] < righthalf[j]:
                arr[k]=lefthalf[i]
                i=i+1
            else:
                arr[k]=righthalf[j]
                j=j+1
            k=k+1

        while i < len(lefthalf):
            arr[k]=lefthalf[i]
            i=i+1
            k=k+1

        while j < len(righthalf):
            arr[k]=righthalf[j]
            j=j+1
            k=k+1
```

In [4]:
```
arr = [11,2,5,4,7,6,8,1,23]
merge_sort(arr)
arr
```

Out[4]: [1, 2, 4, 5, 6, 7, 8, 11, 23]

Good Job!