Divide and Conquer Sorts

Merge Sort

Merge sort recursively divides an array into half, sorts that half, and then merges the two sorted halves back together.

Pseudocode

```
if array.length = base case
  return array

length = arr.length
midIndex = length / 2
left = arr[0:midIndex-1]
right = arr[midIndex:length-1]

merge(left)
merge(right)

initialize i, j
while i and j are not at the end of the left and right arrays:
  if left[i] <= right[i]:
    arr[i+j] = left[i]
    i++
  else:
    array[i+j] = right[j]
    j++

while i < left.length
  arr[i+j] = left[i]
  i++

while j < right.length
  arr[i+j] = right[j]
  j++
```

Theorem 0.1

Time Complexities

For all cases, merge sort is O(nlogn).

Merge sort is stable, but not adaptive and not in place.