

# 1- Merge Sort

July 16, 2019

**Method: Divide & Conquer** Input: an array of unsorted numbers Output: sorted array (increasing) Steps:

1. divide the array into two sub-arrays
2. sort each sub-array recursively
3. merge them back

Run time:  $6 N \log N + 6N$

Implementation: It is a straight forward recursive method, which I implemented via Python v.3.

```
In [68]: import random
import time

#generate a list of integer with random numbers
def generate_list_1():
    size_ = int(input("insert size of the array: "))
    list_1 = [random.randrange(0, 101, 1) for _ in range(size_ + 1)]
    print("\n", "Generated array is: ", "\n", list_1, "\n")
    return list_1

#sorting via MergeSort (Divide&Conquer) technique
def mergesort(list_):
    if (len(list_) > 1):
        len_ = len(list_)
        mid = len_//2
        left = list_[:mid]
        right = list_[mid:]
        #recursive
        mergesort(left)
        mergesort(right)
        i = j = k = 0
        # Copy data to temp arrays: left[] and right[]
        while i < len(left) and j < len(right):
            if left[i] < right[j]:
                list_[k] = left[i]
                i+=1
```

```

        else:
            list_[k] = right[j]
            j+=1
            k+=1
        # Checking if any element was left
        while i < len(left):
            list_[k] = left[i]
            i+=1
            k+=1
        while j < len(right):
            list_[k] = right[j]
            j+=1
            k+=1
    return list_

if __name__ == '__main__':
    start = time.time()
    print("Sorted array is: ", "\n", mergesort(generate_list_1()))
    end = time.time()
    print("time took to run the mergesort is: ", end-start)

```

insert size of the array: 200

Generated array is:

[17, 19, 78, 59, 44, 19, 91, 49, 85, 61, 84, 23, 32, 32, 3, 17, 96, 99, 67, 9, 57, 41, 13, 9,

Sorted array is:

[1, 2, 2, 2, 3, 3, 4, 4, 6, 6, 7, 8, 9, 9, 9, 11, 11, 12, 13, 13, 14, 14, 15, 17, 17, 17, 17,  
time took to run the mergesort is: 1.4646224975585938