MySort.txt

MergeSort Algorithm

1. Definition : This algorithm is divided and conquer algorithm, split into smaller sub arrays and sorting and merging the arrays to make final sorted array.
2. Pseudocode

MergeSort(arr);

If length of arr >1;

mid = length of arr /2

left\_half = arr[0..mid]

right\_half= arr[mid+1..end]

MergeSOrt(left\_half)

MergeSort(right\_half)

Merge(arr, left\_half, right\_half)

Merge(arr, left\_half, right\_half)

Int i=0

Int j =0

Int k =0

while i < length of left\_half and j < length of right\_half:

if left\_half[i] < right\_half[j]:

arr[k] = left\_half[i]

i = i + 1

else:

arr[k] = right\_half[j]

j = j + 1

k = k + 1

while i < length of left\_half:

arr[k] = left\_half[i]

i = i + 1

k = k + 1

while j < length of right\_half:

arr[k] = right\_half[j]

j = j + 1

k = k + 1

1. Complexity Analysis

Time complexity: best and worst cases is O(n Log n)

Space complexity: O(n)