# Merge Sort

- Merge Sort is an *O*(*n* log *n*) comparison-based sorting algorithm.
- Merge Sort is a divide and conquer algorithm.
- It was invented by John von Neumann in 1945.
- A detailed description and analysis of bottom-up merge sort appeared in a report by Goldstine and Neumann as early as 1948.

## Algorithm

Conceptually, a merge sort works as follows:

1. Divide the unsorted list into *n* sub-lists, each containing 1 element (a list of 1 element is considered sorted).
2. Repeatedly merge sub-lists to produce new sorted sub-lists until there is only 1 sub-list remaining. This will be the sorted list.

| | |
|---|---|
| **Worst case performance** | **O(*n* log *n*)** |
| **Best case performance** | **O(*n* log *n*)** |
| **Average case performance** | **O(*n* log *n*)** |
| **Worst case space complexity** | **O(*n*) auxiliary** |

# CODES (C)

```c
#include<stdio.h>

int arr[20];       // array to be sorted

int main()
{
  int n,i;
  printf("Enter the size of array\n");  // input the elements
  scanf("%d",&n);
  printf("Enter the elements:");
  for(i=0; i<n; i++)
    scanf("%d",&arr[i]);
  merge_sort(arr,0,n-1);  // sort the array
  printf("Sorted array:");  // print sorted array
  for(i=0; i<n; i++)
    printf("%d",arr[i]);
  return 0;
}

int merge_sort(int arr[],int low,int high)
{
  int mid;
  if(low<high) {
    mid=(low+high)/2;
    // Divide and Conquer
    merge_sort(arr,low,mid);
    merge_sort(arr,mid+1,high);
    // Combine
    merge(arr,low,mid,high);
  }
  return 0;
}
```

```
int merge(int arr[],int l,int m,int h)
{
  int arr1[10],arr2[10];  // Two temporary arrays to
  hold the two arrays to be merged
  int n1,n2,i,j,k;
  n1=m-l+1;
  n2=h-m;

  for(i=0; i<n1; i++)
    arr1[i]=arr[l+i];
  for(j=0; j<n2; j++)
    arr2[j]=arr[m+j+1];



  arr1[i]=9999;  // To mark the end of each temporary array
  arr2[j]=9999;

  i=0;
  j=0;
  for(k=l; k<=h; k++) { //process of combining two sorted
arrays
    if(arr1[i]<=arr2[j])
      arr[k]=arr1[i++];
    else
      arr[k]=arr2[j++];
  }

  return 0;
}
```

**Code Courtesy:** merge_sort.c hosted with ❤ by GitHub

# CODES (JAVA)

```java
import java.io.*;
import java.util.*;
import java.lang.*;


class MergeSort {

      static public void DoMerge(int []
numbers, int left, int mid, int right)
      {
        int [] temp = new int[25];
        int i, left_end, num_elements, tmp_pos;

        left_end = (mid - 1);
        tmp_pos = left;
        num_elements = (right - left + 1);

        while ((left <= left_end) && (mid <= right))
        {
            if (numbers[left] <= numbers[mid])
                temp[tmp_pos++] = numbers[left++];
            else
                temp[tmp_pos++] = numbers[mid++];
        }

        while (left <= left_end)
            temp[tmp_pos++] = numbers[left++];

        while (mid <= right)
            temp[tmp_pos++] = numbers[mid++];

        for (i = 0; i < num_elements; i++)
        {
            numbers[right] = temp[right];
            right--;
        }
    }
    static public void MergeSort_Recursive(int []
numbers, int left, intright)
    {
      int mid;

      if (right > left)
      {
        mid = (right + left) / 2;
        MergeSort_Recursive(numbers, left, mid);
        MergeSort_Recursive(numbers, (mid + 1), right);

        DoMerge(numbers, left, (mid+1), right);
      }
    }


    public static void main(String[] args)
      {
        int[] numbers = { 47, 18, 23, 19, 1, 72, 87, 44, 21 };
        int len = 9;

        System.out.println("Output:");

        MergeSort_Recursive(numbers, 0, len - 1);
        for (int i = 0; i < 9; i++)
            System.out.println(numbers[i]);

    }
}
```

Code Courtesy: http://www.softwareandfinance.com/