Sorting Algorithms:

Bubble sort:

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

    n = len(arr)-1

    for i in range(n):

        for j in range(0, n-i):

            if arr[j]>arr[j+1]:
                temp = arr[j]
                arr[j] = arr[j+1]
                arr[j+1] = temp

    return arr
```

In [4]:
```
arr = [45,89,12,75,106,5]
bubble_sort(arr)
```

Out[4]:
```
[5, 12, 45, 75, 89, 106]
```

In [5]:
```
arr2 = [23,1,45,56,12,34,44,11,10,9]
bubble_sort(arr2)
```

Out[5]:
```
[1, 9, 10, 11, 12, 23, 34, 44, 45, 56]
```

Selection Sort:

In [17]:
```
def selec_sort(arr):

    for i in range(0,len(arr)):

        min_idx = i

        for j in range(i+1,len(arr)):

            if arr[min_idx]>arr[j]:

                min_idx = j

        arr[i],arr[min_idx] = arr[min_idx],arr[i]

    return arr
```

In [18]:
```
arr1 = [34,1,23,59,21,78,32]
selec_sort(arr1)
```

Out[18]:
```
[1, 21, 23, 32, 34, 59, 78]
```

In [19]:
```
def selection_sort(L):

    for i in range(len(L)):

        min_index = i

        for j in range(i+1, len(L)):

            if L[j] < L[min_index]:
                min_index = j

        temp = L[i]
        L[i] = L[min_index]
        L[min_index] = temp

    return L
```

In [20]:
```
arr1 = [34,1,23,59,21,78,32]
selection_sort(arr1)
```

Out[20]:
```
[1, 21, 23, 32, 34, 59, 78]
```

Insertion Sort:

In [12]:
```
def insertion_sort(arr):

    for i in range(1,len(arr)):

        current_value = arr[i]
        position = i

        while position>0 and arr[position-1]>current_value:

            arr[position] = arr[position-1]
            position = position-1

        arr[position] = current_value

    return arr
```

In [13]:
```
arr = [50,30,10,80,20,40]
```

In [14]:
```
insertion_sort(arr)
```

Out[14]:
```
[10, 20, 30, 40, 50, 80]
```

Merge Sort:

In [29]:
```
def merge_sort(arr):

    if len(arr)>1:

        mid = int(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 +=1

            else:

                arr[k] = righthalf[j]
                j +=1

            k +=1

        while i<len(lefthalf):

            arr[k] = lefthalf[i]

            i +=1
            k +=1

        while j<len(righthalf):

            arr[k] = righthalf[j]

            j +=1
            k +=1

    print('Merging : ',arr)
    return arr
```

In [30]:
```
arr = [34,6,2,68,1,7,4,7,21]
merge_sort(arr)
```

```
Merging :  [34]
Merging :  [6]
Merging :  [6, 34]
Merging :  [2]
Merging :  [68]
Merging :  [2, 68]
Merging :  [2, 6, 34, 68]
Merging :  [1]
Merging :  [7]
Merging :  [1, 7]
Merging :  [4]
Merging :  [7]
Merging :  [21]
Merging :  [7, 21]
Merging :  [4, 7, 21]
Merging :  [1, 4, 7, 7, 21]
Merging :  [1, 2, 4, 6, 7, 7, 21, 34, 68]
```

Out[30]:
```
[1, 2, 4, 6, 7, 7, 21, 34, 68]
```

Quick Sort:

In [31]:

```
def quick_sort(arr):
    quick_sort_help(arr, 0, len(arr)-1)

def quick_sort_help(arr, first, last):

    if first<last:

        splitpoint = partition(arr, first, last)

        quick_sort_help(arr,first, splitpoint-1)
        quick_sort_help(arr,splitpoint+1,last)

def partition(arr, first, last):

    pivotvalue = arr[first]

    leftmark = first+1
```

In [ ]: