Quick Sort

- Also called as Partition Exchange Sort
- Divide and Conquer Algorthm Approach

1. Divide
2. Conquer
3. Combine

In [7]:

```
# program to implement quick sort
# To get the correct position of the pivot element
# In this Example I am Taking pivot as last
def pivot_place(a,first,last):
    pivot=a[last]
    left=first
    right=last-1
    while True:
        while left<=right and a[left]<=pivot:
            left=left+1
        while left<=right and a[right]>=pivot:
            right=right-1
        if right<left:
            break

        else:
            a[left],a[right]=a[right],a[left]
    a[last],a[left]=a[left],a[last]
    return left
# To Divide the list
def quicksort(a,first,last):
    if first<last:
        p=pivot_place(a,first,last)
        quicksort(a,first,p-1)
        quicksort(a,p+1,last)
# User Input
a=[14,25,100,14,1,17]
print("Before Sorting",a)
n=len(a)
quicksort(a,0,n-1)
print("After Sorting",a)
```

```
Before Sorting [14, 25, 100, 14, 1, 17]
After Sorting [1, 14, 14, 17, 25, 100]
```

In [5]:

```
# I am taking pivot as First element
def pivot_place(a,first,last):
    pivot=a[first]
    left=first+1
    right=last
    while True:
        while left<=right and a[left]<=pivot:
            left=left+1
        while left<=right and a[right]>=pivot:
            right=right-1
        if right<left:
            break

        else:
            a[left],a[right]=a[right],a[left]
    a[first],a[right]=a[right],a[first]
    return right
# To Divide the list
def quicksort(a,first,last):
    if first<last:
        p=pivot_place(a,first,last)
        quicksort(a,first,p-1)
        quicksort(a,p+1,last)
# User Input
a=[14,25,100,14,1,17]
print("Before Sorting",a)
n=len(a)
quicksort(a,0,n-1)
print("After Sorting",a)
```

```
Before Sorting [14, 25, 100, 14, 1, 17]
After Sorting [1, 14, 14, 17, 25, 100]
```

In [6]:

```
# I am going to take random element as my pivot element
import random

def pivot_place(a,first,last):
    rand_int=random.randint(first,last)
    a[rand_int],a[first]=a[first],a[rand_int]
    pivot=a[first]
    left=first+1
    right=last
    while True:
        while left<=right and a[left]<=pivot:
            left=left+1
        while left<=right and a[right]>=pivot:
            right=right-1
        if right<left:
            break

        else:
            a[left],a[right]=a[right],a[left]
    a[first],a[right]=a[right],a[first]
    return right
# To Divide the list
def quicksort(a,first,last):
    if first<last:
        p=pivot_place(a,first,last)
        quicksort(a,first,p-1)
        quicksort(a,p+1,last)
# User Input
a=[14,25,100,14,1,17]
print("Before Sorting",a)
n=len(a)
quicksort(a,0,n-1)
print("After Sorting",a)
```

```
Before Sorting [14, 25, 100, 14, 1, 17]
After Sorting [1, 14, 14, 17, 25, 100]
```

Merge Sort

Algorithm:

- 1) Split the Unsorted list
- 2) Compare each of the element and group them
- 3) Repeat step2 untill whole list is merged and Sorted

In [2]:

```
# Write a Merge Sort program to sort the list of elements in ascending order
def Mergesort(lst):
    # Dividing the list of elements into single elements
    if len(lst)>1:
        mid=len(lst)//2
        l_sub=lst[:mid]
        r_sub=lst[mid:]
        Mergesort(l_sub)
        Mergesort(r_sub)
    # Merging the elements
        i=0
        j=0
        k=0
        while i<len(l_sub) and j<len(r_sub):
            if l_sub[i]<r_sub[j]:
                lst[k]=l_sub[i]
                i+=1
                k+=1
            else:
                lst[k]=r_sub[j]
                j+=1
                k+=1
    # If any element is left in the left sublist we can add
        while i<len(l_sub):
            lst[k]=l_sub[i]
            i+=1
            k+=1
     # If any element is left in the right sublist we can add
        while j<len(r_sub):
            lst[k]=r_sub[j]
            j+=1
            k+=1
# main part(User Input)
num=int(input("Enter How many elements you want in list:"))
lst=[int(input("Enter element:")) for x in range(num)]
print("Before Sorting The Elements:",lst)
Mergesort(lst)
print("After Sorting The Elements :",lst)
```

```
Enter How many elements you want in list:7
Enter element:19
Enter element:2
Enter element:60
Enter element:40
Enter element:11
Enter element:7
Enter element:18
Before Sorting The Elements: [19, 2, 60, 40, 11, 7, 18]
After Sorting The Elements : [2, 7, 11, 18, 19, 40, 60]
```

```
In [ ]:
```