Implementation of Quick Sort

A quick sort first selects a value, which is called the pivot value. Although there are many different ways to choose the pivot value, we will simply use the first item in the list. The role of the pivot value is to assist with splitting the list. The actual position where the pivot value belongs in the final sorted list, commonly called the split point, will be used to divide the list for subsequent calls to the quick sort.

Resources for Review

Check out the resources below for a review of Insertion sort!

- Wikipedia
- Visual Algo
- Sorting Algorithms Animcation with Pseudocode

In [1]:

```
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
    rightmark = last

    done = False
    while not done:

        while leftmark <= rightmark and arr[leftmark] <= pivotvalue:
            leftmark = leftmark + 1

        while arr[rightmark] >= pivotvalue and rightmark >= leftmark:
            rightmark = rightmark -1

        if rightmark < leftmark:
            done = True
        else:
            temp = arr[leftmark]
            arr[leftmark] = arr[rightmark]
            arr[rightmark] = temp
    temp = arr[first]
    arr[first] = arr[rightmark]
    arr[rightmark] = temp


    return rightmark
```

In [2]:

```
arr = [2,5,4,6,7,3,1,4,12,11]
quick_sort(arr)
arr
```

Out[2]:

```
[1, 2, 3, 4, 4, 5, 6, 7, 11, 12]
```

Good Job!