Quick sort Fix pivot in Python

```
# function to find the partition position
def partition(array, low, high):
    # choose the rightmost element as pivot
    pivot = array[high]
    # print(pivot, end=" ")

    # pointer for greater element
    i = low - 1

    # traverse through all elements
    # compare each element with pivot
    for j in range(low, high):
        if array[j] <= pivot:
            # if element smaller than pivot is found
            # swap it with the greater element pointed by i
            i = i + 1

            # swapping element at i with element at j
            (array[i], array[j]) = (array[j], array[i])

    # swap the pivot element with the greater element specified by i
    (array[i + 1], array[high]) = (array[high], array[i + 1])

    # return the position from where partition is done
    return i + 1


# function to perform quicksort
def quickSort(array, low, high):
    if low < high:
        # find pivot element such that
        # element smaller than pivot are on the left
        # element greater than pivot are on the right
        pi = partition(array, low, high)

        # recursive call on the left of pivot
        quickSort(array, low, pi - 1)

        # recursive call on the right of pivot
        quickSort(array, pi + 1, high)


data = list(map(int, input("Enter elements of array:-").strip().split()))

print("Unsorted Array (Pivot last element of array)")
```

```
print(data, "\n")

size = len(data)

quickSort(data, 0, size - 1)

print('\nSorted Array in Ascending Order:')
print(data)
"""
Time Complexity
Best   O(n*log n)
Worst  O(n^2)
Average    O(n*log n)
Space Complexity   O(log n)
Stability  No
"""
```

Output:

Enter elements of array: -6 10 5 3 8 7 9

Unsorted Array (Pivot last element of array)

[6, 10, 5, 3, 8, 7, 9]

Sorted Array in Ascending Order:

[3, 5, 6, 7, 8, 9, 10]

```
# Python implementation QuickSort using
# Lomuto's Random partition Scheme.
import random

'''
The function which implements QuickSort.
arr :- array to be sorted.
start :- starting index of the array.
stop :- ending index of the array.
'''

def quicksort(arr, start, stop):
    if start < stop:
        # pivotindex is the index where
        # the pivot lies in the array
        pivotindex = partitionrand(arr, start, stop)

        # At this stage the array is
        # partially sorted around the pivot.
        # Separately sorting the
        # left half of the array and the
        # right half of the array.
        quicksort(arr, start, pivotindex - 1)
        quicksort(arr, pivotindex + 1, stop)


# This function generates random pivot,
# swaps the first element with the pivot
# and calls the partition function.
def partitionrand(arr, start, stop):
    # Generating a random number between the
    # starting index of the array and the
    # ending index of the array.
    randpivot = random.randrange(start, stop)

    # Swapping the starting element of
    # the array and the pivot
    arr[start], arr[randpivot] = \
        arr[randpivot], arr[start]
    return partition(arr, start, stop)

'''
This function takes the first element as pivot,
places the pivot element at the correct position
in the sorted array. All the elements are re-arranged
according to the pivot, the elements smaller than the
pivot is places on the left and the elements
```

```
greater than the pivot is placed to the right of pivot.
'''

def partition(arr, start, stop):
    pivot = start  # pivot

    # a variable to memorize where the
    i = start + 1

    # partition in the array starts from.
    for j in range(start + 1, stop + 1):

        # if the current element is smaller
        # or equal to pivot, shift it to the
        # left side of the partition.
        if arr[j] <= arr[pivot]:
            arr[i], arr[j] = arr[j], arr[i]
            i = i + 1
    arr[pivot], arr[i - 1] = \
        arr[i - 1], arr[pivot]
    pivot = i - 1
    return pivot

# Driver Code
if __name__ == "__main__":
    data = list(map(int, input("Enter elements of array:-").strip().split()))

    print("Unsorted Array (Pivot Random element of array)")
    print(data, "\n")
    quicksort(data, 0, len(data) - 1)
    print('\nSorted Array in Ascending Order:')
    print(data)
```

Output:

Enter elements of array:-6 10 5 3 8 7 9

Unsorted Array (Pivot Random element of array)

[6, 10, 5, 3, 8, 7, 9]

Sorted Array in Ascending Order:

[3, 5, 6, 7, 8, 9, 10]