Name : Yash Srivastava || Section - A || Roll. No - 63

Implementation and Analysis of Quick Sort

```
def partition(arr, l, h):
    i = (l - 1)
    pivot = arr[h]
    for j in range(l, h):
        if arr[j] < pivot:
            i = i + 1
            arr[i], arr[j] = arr[j], arr[i]

    arr[i + 1], arr[h] = arr[h], arr[i + 1]
    return (i + 1)


def quickSort(arr, l, h):
    if l < h:
        p = partition(arr, l, h)
        quickSort(arr, l, p - 1)
        quickSort(arr, p + 1, h)


arr = list(map(int,input("Enter values for Array :
\n").split()))
n = len(arr)
quickSort(arr, 0, n - 1)
print("Sorted array is:")
for i in range(n):
    print(arr[i])
```

Input:

10 50 20 30 60

Name : Yash Srivastava || Section - A || Roll. No - 63

Output:

10 20 30 50 60

Time Complexity:

- O(nlogn) average and best
- O(n^2) worst

Program:

Quick Sort.py

```
def partition(arr, l, h):
    i = (l - 1)
    pivot = arr[h]
    for j in range(l, h):
        if arr[j] < pivot:
            i = i + 1
            arr[i], arr[j] = arr[j], arr[i]

    arr[i + 1], arr[h] = arr[h], arr[i + 1]
    return (i + 1)


def quickSort(arr, l, h):
    if l < h:
        p = partition(arr, l, h)
        quickSort(arr, l, p - 1)
        quickSort(arr, p + 1, h)


arr = list(map(int,input("Enter values for Array : \n").split()))
n = len(arr)
quickSort(arr, 0, n - 1)
print("Sorted array is:")
for i in range(n):
    print(arr[i])
```

Input/Output:

```
Quick Sort
C:\Anaconda\envs\MLProject\python.exe "C:\MLProject\Quick Sort.py"
Enter values for Array :
10 50 20 30 60
Sorted array is:
10
20
30
50
60

Process finished with exit code 0
```

TODO 4: Run Terminal Python Console Event Log

rm 2020.1.4 available: // Update... (today 13:30) 21:13 CRLF UTF-8 4 spaces Python 3.6 (MLProject)