Lab report of CSE 206

Course Title: Algorithms Lab
Course Code: CSE 206
Section: PC DA

Submitted to
Dr. Shah Murtaza Rashid Al Masud
Assistant Professor
Dept. of CSE
Green University of Bangladesh

Submitted by:
Mohammad Nazmul Hossain
ID:193902031
Dept. of CSE

Question: Explain Quicksort algorithm with example

Answer to the Q:-

Code in JavaScript:-

Quick Sort Algorithm

Quicksort is one of the most efficient ways of sorting elements in computer systems. Similor to merge sort, Quicksort works on the divide and conquer algorithm. In this, we find a pivot item in the array to compare all other elements arrays against and then we move the items in a way where all items before our selected pivot items are smaller and all items after the pivot item are larger in size. Once we have done that, the key is to keep doing it repeatedly and we will have our sorted array.

Following are the steps that can be followed to implement the quicksort algorithm:

- ★ We select an element of the array and call it “Pivot Point”
- ★ We start a pointer called the left pointer from which is at the first element in the array.
- ★ Similarly, we start a pointer called the right pointer at the last item in the array.
- ★ If the value of the element at the left pointer is less compared to the selected pivot point, we move the left pointer leftwards (add +1 to it) and keep repeating it until the value at the left pointer is found to be bigger than the value of pivot point or equal to it.
- ★ If the value of the element at the right pointer in the list is higher than the value of the pivot element, we mode the right pointer to left. Repeat this until the value at the right side pointer is lower than (or equal to) the value of pivot.

★ When the value of the left pointer is less than or equal to the value of the right pointer, swap the values.

★ Move the right pointer to left by one, left pointer to right by one.

★ Repeat until left and right pointers meet.

Code:

```
function quickSortAlgo(origArray) {
if (origArray.length <= 1) {
return origArray;
} else {
var left = [];
var right = [];
var newArray = [];
var pivot = origArray.pop();
var length = origArray.length;
for (var i = 0; i < length; i++) {
if (origArray[i] <= pivot) {
left.push(origArray[i]);
} else {
```

```
right.push(origArray[i]);
}
}
return newArray.concat(quickSortAlgo(left), pivot, quickSortAlgo(right));
}
}
var myArray = [13, 50, 2, 45, -1, 74, 11 ];
var arreySorted = quickSortAlgo(myArray);
console.log(arreySorted);
```

Output:

```
[-1,2,11,13,45,50,74]
```