RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Lab report: 03
Course No.: CSE 2202

Date of Experiment: 13.11.2018
Date of Submission: 20.11.2018

Submitted to:
Biprodip Pal
Assistant Professor,
Department of Computer
Science and Engineering
Rajshahi University of
Engineering and Technology

Submitted by:
Riyad Morshed Shoeb
Roll No: 1603013
Section: A
Department of Computer
Science and Engineering
Rajshahi University of
Engineering and Technology

Problem: Divide and conquer sorting approach to Quick Sort.

Approach-1:

Code:

```cpp
#include <cstdio>
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
void swap(int* a, int* b)
{
  int temp;
  temp = *a;
  *a = *b;
  *b = temp;
}
int FindPosition(int array[], int first, int last)
{
  int pivot = array[first];
  int position = last+1;
  int iterator;
  for(iterator=last; iterator>first; iterator--)
  {
    if(pivot <= array[iterator])
    {
      position--;
      swap(&array[iterator], &array[position]);
    }
  }
  swap(&array[first], &array[position-1]);
  return position;
}
int main(void)
{
  int INPUT_SIZE;
  int iterator;
  int PIVOT_POSITION;
  cout << "Enter size of the input: ";
  cin>>INPUT_SIZE;
  int data[INPUT_SIZE];
  srand(time(0));
  for(iterator=0; iterator<INPUT_SIZE; iterator++)
  {
    data[iterator]= rand()%1001;
    cout << data[iterator] << endl;
  }
  cout << endl;
  PIVOT_POSITION = FindPosition(data, 0, INPUT_SIZE-1);
  cout<<"Position: "<< PIVOT_POSITION << endl;
  for(iterator=0; iterator<INPUT_SIZE; iterator++)
    cout << data[iterator] << endl;
}
```

Output:

```
C:\ pivoting
Enter size of the input: 10
276
682
106
603
665
209
340
577
631
128
Position: 4
209
128
106
276
682
603
665
340
577
631

Press any key to continue . . . ▄
```

Approach-2:

Code:

```cpp
#include <cstdio>
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
void swap(int* a, int* b)
{
  int temp;
  temp = *a;
  *a = *b;
  *b = temp;
}
int FindPosition(int array[], int first, int last)
{
  int pivot = array[first];
  int position = last+1;
  int iterator;
  for(iterator=last; iterator>first; iterator--)
  {
    if(pivot <= array[iterator])
    {
      position--;
      swap(&array[iterator], &array[position]);
    }
  }
  swap(&array[first], &array[position-1]);
  return position-1;
}
void QuickSort(int array[], int first_index, int last_index)
```

```
{
  int PIVOT_POSITION;
  PIVOT_POSITION = FindPosition(array, first_index, last_index);
  if(PIVOT_POSITION > first_index+1)
    QuickSort(array, first_index, PIVOT_POSITION-1);
  if(PIVOT_POSITION < last_index-1)
    QuickSort(array, PIVOT_POSITION+1, last_index);
}
int main(void)
{
  int INPUT_SIZE;
  int iterator;
  cout << "Enter size of the input: ";
  cin>>INPUT_SIZE;
  int data[INPUT_SIZE];
  srand(time(0));
  for(iterator=0; iterator<INPUT_SIZE; iterator++)
  {
    data[iterator]= rand()%1001;
    cout << data[iterator] << endl;
  }
  QuickSort(data, 0, INPUT_SIZE-1);

  cout << "After sorting:" << endl;
  for(iterator=0; iterator<INPUT_SIZE; iterator++)
    cout << data[iterator] << endl;
}
```

Output:

```
sort
Enter size of the input: 10
288
703
724
967
370
178
805
770
734
406
After sorting:
178
288
370
406
703
724
734
770
805
967

Press any key to continue . . .
```