

Name : Lakhan Kumawat.

Roll No: 1906055.

Course: CSL4403.

Design and Analysis of Algorithms Lab

## Lab 02( Que 1)

Write a program to implement Quick sort algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#define pf printf
#define sf scanf

//Time complexity best case and average case  $O(n(\log n))$  worst case when elements are sorted is  $O(n^2)$ 
//Space required is constant void
QuickSort(int a[],int low,int high){ int
pivot;
// Termination condition
if(high>low){ pivot=
Partition(a,low,high);
QuickSort(a,low,pivot-1);
QuickSort(a,pivot+1,high);
}
}

int Partition(int a[],int low,int high){
int left,right,pivot_item= a[low];
left=low; right=high;
while(left<right){
//Move left while item left is pointing is <=pivot
while(a[left]<=pivot_item) left++;
while(a[right]>pivot_item) right--; if(left<right){
//swap indexes of left and right
int temp=a[left]; a[left]=a[right];
a[right]=temp;
}
}
//Final position of right is of the pivot
a[low]=a[right]; a[right]=pivot_item;
//Return the index of right element
return right;
}
```

```

void display(int a[],int size){

for(int i=0;i<size;i++)
    pf("%d\n",a[i]);
}
int main()
{   int
no,i;
    pf("Enter the Elements you want to sort : \n");
sf("%d",&no);  int Ele[no];  for(i=0;i<no; i++)
    sf("%d",&Ele[i]);

    QuickSort(Ele,0,no-1);pf("\nQuickSort Applied\n");
    display(Ele,no);
}

```

## Outputs :

```

C:\Users\Lakhan Kumawat\Documents\DataStructureCodeFiles\QueueAndRest\bin\Debug\QueueAndRest.exe
Enter the Elements you want to sort :
5
9
1
5
6
2

QuickSort Applied
1
2
5
6
9

Process returned 0 (0x0)   execution time : 16.860 s
Press any key to continue.

```

```

C:\Users\Lakhan Kumawat\Documents\DataStructureCodeFiles\QueueAndRest\bin\Debug\QueueAndRest.exe
Enter the Elements you want to sort :
4
91
1
8
1

QuickSort Applied
1
1
8
91

Process returned 0 (0x0)   execution time : 15.927 s
Press any key to continue.

```

## Que 2

Write a program to implement Binary Search.

```
#include<stdio.h>
#include<stdlib.h>
#define pf printf
#define sf scanf
int isEven(int n){
return(n%2==0);}

void BinarySearch(int a[],int n,int desired){
    int first = 0;
    int last = n-1;
    int mid = (first +last)/2;
    while(first<=last){
        if(a[mid]<desired)
            first= mid+1;
        else if(a[mid]==desired){
            pf("%d found at index %d",desired,mid+1);
            break;}
        else
            last = mid -1;

        mid= (first +last)/2;
    }
    if(first>last)
        pf("Not Found ! %d is not present in the List..",desired);
    }

int main()
{
    int no,i,choice,s;
    pf("Enter the total No. of Elements : \n");
    sf("%d",&no);
    int Ele[no];
    for(i=0;i<no; i++)
        sf("%d",&Ele[i]);
    pf("Enter the Element you want to perform Binary Search : \n");
    sf("%d",&s);

    BinarySearch(Ele,no,s);
}
```

**Outputs :**

## 1. Data is sorted .

```
"C:\Users\Lakhan Kumawat\Documents\DataStructureCodeFiles\QueueAndRest\bin\Debug\QueueAndRest.exe"
Enter the total No. of Elements :
5
1
9
15
19
23
Enter the Element you want to perform Binary Search :
19
19 found at index 4
Process returned 0 (0x0)   execution time : 15.309 s
Press any key to continue.
```

## 2. Data is Not sorted.

```
"C:\Users\Lakhan Kumawat\Documents\DataStructureCodeFiles\QueueAndRest\bin\Debug\QueueAndRest.exe"
Enter the total No. of Elements :
4
11
13
9
20
Enter the Element you want to perform Binary Search :
9
Not Found ! 9 is not present in the List..
Process returned 0 (0x0)   execution time : 25.469 s
Press any key to continue.
```