Design and Analysis of Algorithms – 20ISL57A

Program 1 - Implement and analyze quick sort algorithm

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int partition(int a[], int low, int high)
{
   int pivot=a[low], i=low, j=high+1;
   int temp;
   while(i<j)
   {
           do
           {
              i++;
           }while(pivot>=a[i] && i<high);
           do
           {
              j--;
           }while(pivot<a[j]);
           if(i<j)
           {
             temp=a[i];
             a[i]=a[j];
             a[j]=temp;
           }
   }
   a[low]=a[j];
   a[j]=pivot;
   return j;
}
```

```
void quick_sort(int a[], int low, int high)
{
   int s;
   if(low<high)
   {
          s=partition(a,low,high);
          quick_sort(a,low,s-1);
          quick_sort(a,s+1,high);
   }
}
int main()
{
   int a[10000],n,low,high,i;
   clock_t st, end;
   printf("Enter number of elements\n");
   scanf("%d",&n);
   printf("Random numbers generated are\n");
   for(i=0;i<n;i++)
   {
          a[i]=rand()%100;
          printf("%d\t",a[i]);
   }
   low=0;
   high=n-1;
   st=clock();
   quick_sort(a,low,high);
   end=clock();
   printf("\nSorted array\n");
   for(i=0;i<n;i++)
   {
          printf("%d\t",a[i]);
   }
   printf("\nTime required to sort given elements is %f",(float)(end-st)/CLOCKS_PER_SEC);
}
```