

10) Design and implement C/C++ Program to sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n > 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Soln:

```
#include<stdio.h>

#include<stdlib.h>

#include<time.h>

void swap(int* xp,int* vp)

{

    int temp=*xp;

    *xp=*vp;

    *vp=temp;

}

int partition(int arr[],int low,int high)

{

    int pivot=arr[high];

    int i=low-1;

    int j;

    for(j=low;j<=high-1;j++)

    {

        if(arr[j]<=pivot)

        {

            i++;

            swap(&arr[i],&arr[j]);

        }

    }

}
```

```

    swap(&arr[i+1],&arr[high]);

    return(i+1);
}

void quickSort(int arr[],int low,int high)

{
    if(low<high)
    {
        int pi=partition(arr,low,high);

        quickSort(arr,low,pi-1);

        quickSort(arr,pi+1,high);
    }
}

void genrandm(int arr[],int n)

{
    int i;

    for(i=0;i<n;i++)
    {
        arr[i]=rand()%n;
    }
}

int main()

{
    int n;

    double cpu_time_used,tottime=0

    ;

    for(n=5000;n<=100000;n+=5000)
    {

```

```
int* arr=(int*)malloc(n*sizeof(int));

genrandm(arr,n);

clock_t start,end;


start=clock();

quickSort(arr,0,n-1);

end=clock();

cpu_time_used=((double)(end-start))/CLOCKS_PER_SEC;

printf("time taken to sort %d elements:%f\n",n,cpu_time_used);

tottime= tottime+cpu_time_used;

free(arr);

}

printf("total execution time =%f sec",tottime);//

//use this above line if you are running this line in vs code,else if you are running this code in codeblocks,it doesn't
required// .

return 0;

}
```