Name: Sakshi D Lonare
UID: 2021300069
Class: SE Comps A

Aim:
Each student have to generate random 100000 numbers using rand() function and use this input as 10 blocks of 10000 integer numbers to Merge sort algo.

Algorithm:

1. Start
2. Create an array of length 100000.
3. Input 100000 random integers into both the arrays using : rand()%100000
4. Store the generated random numbers in a text file.
5. Perform insertion sort and selection sort of all the elements in groups of 10000, then 20000, ..so on till end.
6. Print the time taken for each sorting using clock() function.
7. Stop
8. Step 1: Start
   Step 2: Declare an array and left, right, mid variable
   Step 3: Perform merge function.
   mergesort(array,left,right)
   mergesort (array, left, right)
   if left > right
   return
   mid= (left+right)/2
   mergesort(array, left, mid)
   mergesort(array, mid+1, right)
   merge(array, left, mid, right)
   Step 4: Stop

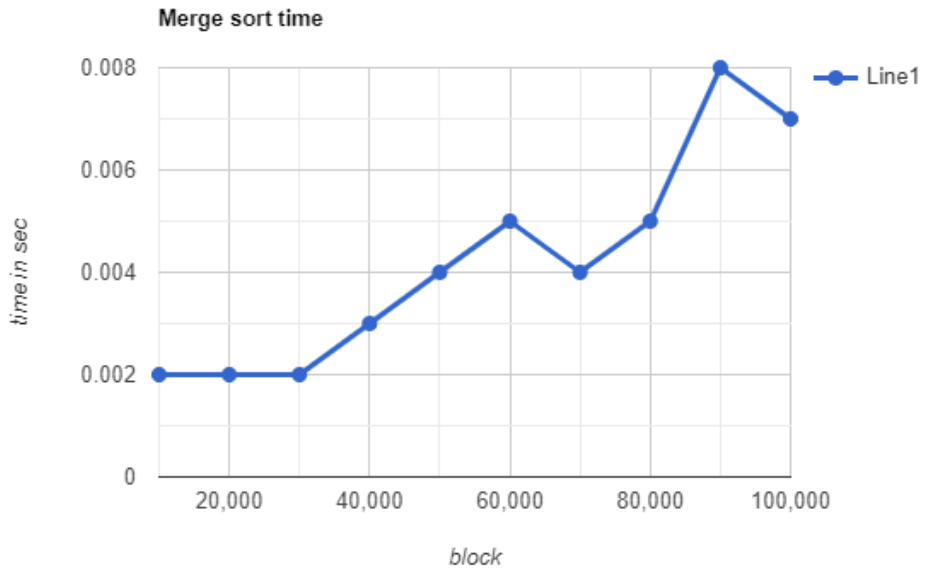
Program:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];
```

```
    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int m = l + (r - l) / 2;

        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}

int main(){
FILE *fptr;
FILE *ansdoc;
```

```
FILE *timedoc;
clock_t start, end;
fptr = fopen("demo.txt","w");
ansdoc = fopen("ans.txt","w");
timedoc = fopen("time.txt","w");

int arr[100000];
for(int i=0;i<100000;i++){
int x=rand()%10000;
printf("%d\n",x);
fprintf(fptr,"%d\n",x);
arr[i]=x;
}

for(int i=1;i<=10;i++)
{
start = clock();
mergeSort(arr,0,i*10000);
end = clock();
double time_taken = (double)(end - start) / (double)(CLOCKS_PER_SEC);
printf("%fs\n",time_taken);
fprintf(timedoc,"%f\n",time_taken);
}

for(int i=1;i<=10000;i++){
fprintf(ansdoc,"Sorted arr %d\n",i);
for(int j=0;j<i*10000;j++){
fprintf(ansdoc,"%d\n",arr[j]);
}
}
fclose(fptr);
fclose(ansdoc);
fclose(timedoc);

return 0;
}
```

Graph & observation:



Conclusion:

After performing the above experiment, I got to know how to find running time of any function in C using the clock() function. Also I learnt two sorting algorithms, namely insertion sort and selection sort.