NAME:- Sourav Paul
GROUP:- D2
REG. NO:- 20214056
BRANCH:- CSE DEPT.

ASSIGNMENT - 04

1). Write a C Program to analyse the time complexity of Quick Sort Algorithm. Also plot its graph for all cases.

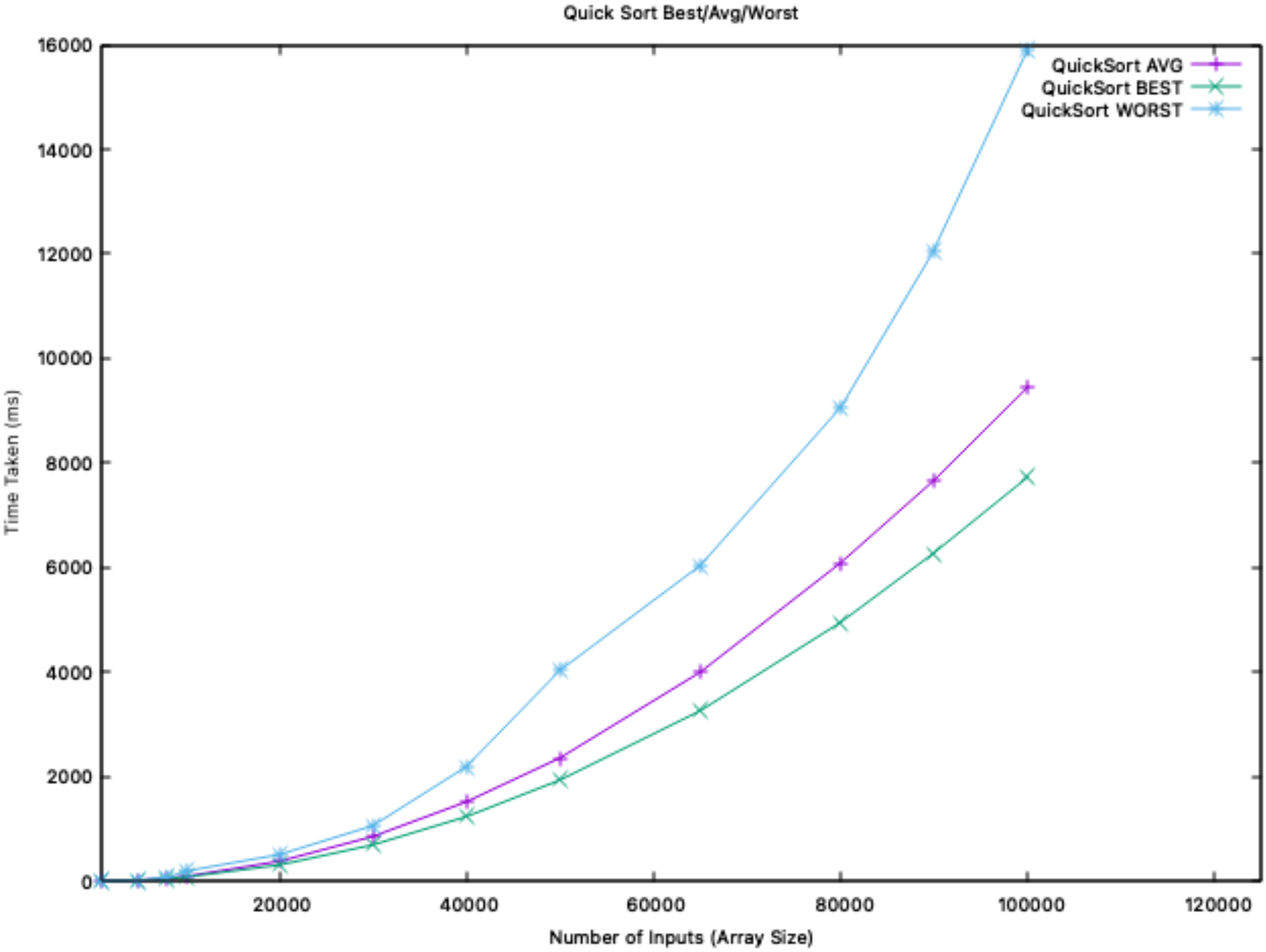
Ans:-

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

clock_t begin;
clock_t end;

void swp(int *x, int *y) {
   int t = *x;
   *x = *y;
   *y = t;
}

int divArr(int arr[], int low, int high)
{
```

```
{
        int pivot = arr[high];
        int i = (low - 1), j;

        for (j = low; j <= high - 1; j++) {
                if (arr[j] < pivot) {
                        i++;
                        swp(&arr[i], &arr[j]);
                }
        }
        swp(&arr[i + 1], &arr[high]);
        return (i + 1);
}

void qSort(int arr[], int low, int high)
{
        if (low < high) {
                int pi = divArr(arr, low, high);
                qSort(arr, low, pi - 1);
                qSort(arr, pi + 1, high);
        }
}

void writeTable(int size, double time, char *filename) {
   int i;
   FILE *fp = fopen(filename, "a+");

   if (fp == NULL) printf("FILE CANNOT BE OPENED\n");
   else {
      fprintf(fp, "%d %lf", size, time);
      fprintf(fp, "\n");
   }
   fclose(fp);
}

void readData(int arr[], char *filename) {
   FILE *fp = fopen(filename, "r+");
```

```
    char x[16];
    int i, k = 0;

    if (fp == NULL) printf("FILE CANNOT BE OPENED\n");
    else {
        while (fgets(x, 16, fp) != NULL) {
            int num = 0;
            fscanf(fp, "%d", &num);
            if (num == 0) break;
            arr[k++] = num;
        }
    }
    fclose(fp);
}

int main(int argc, char **argv) {
    int arr[100000];
    int size[] = {1000, 5000, 8000, 10000, 20000, 30000, 40000,
50000, 65000, 80000, 90000, 100000};
    int i = 0;
    for (i = 0; i <= 11; i++) {
        readData(arr, argv[1]);
        begin = clock();
        qSort(arr, 0, size[i]-1);
        end = clock();
        writeTable(size[i], (end - begin) / 2000.0, argv[2]);
    }
    return 0;
}
```

```
set autoscale                    # scale axes automatically
unset log                        # remove any log-scaling
unset label                      # remove any previous labels
set xtic auto                    # set xtics automatically
set ytic auto                    # set ytics automatically         set tics
font "Helvetica,10"
set title "Quick Sort Best/Avg/Worst"
set xlabel "Number of Inputs (Array Size)"
set ylabel "Time Taken (ms)"

#set key 0.01,100
#set label "Yield Point" at 0.003,260
#set arrow from 0.0028,250 to 0.003,280

set xr [1000:125000]
set yr [0:7500]
plot    "qTableAVG.txt" using 1:2 title 'QuickSort AVG' with
linespoints, \
        "qTableBST.txt" using 1:2 title 'QuickSort BEST' with
linespoints, \
        "qTableWST.txt" using 1:2 title 'QuickSort WORST' with
linespoints
```


Quick Sort Best/Avg/Worst
QuickSort AVG
QuickSort BEST
QuickSort WORST
Time Taken (ms)
0
2000
4000
6000
8000
10000
12000
14000
16000
20000
40000
60000
80000
100000
120000
Number of Inputs (Array Size)