

Student Name:-Srishti Shrivastava

Student Roll No.:-1905647

Algorithm Lab. Class Assignment-4

CSE Group 1

Date: - 30th July 2021

1. Write a C program for bubble sort to

- I. Compare the time complexity with the given data set given below and calculate the time complexity based on the CPU clock.**
- II. Plot a graph showing the comparison (n, the input data Vs. CPU times)**

Sl No.	Value of n	Bubble Sort (Time Complexity)		
		Best case	Average case	Worst case
1	5000	0.041745	0.111071	0.150295
2	10000	0.190776	0.437938	0.268839
3	15000	0.328395	0.873612	1.097408
4	20000	0.719513	1.946363	2.381139
5	25000	0.952250	2.215002	1.774865
6	30000	1.444239	3.328650	2.375833
7	35000	1.780408	4.428513	3.104397
8	40000	2.563681	5.918491	4.161112
9	45000	2.876256	7.690537	5.417063
10	50000	3.349706	8.821329	6.564277

Program

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

void bubbleSort(int a[], int number) {
    int i, j, temp;
    for(i = 0; i < number - 1; i++) {
        for(j = 0; j < number - i - 1; j++) {
            if(a[j] > a[j + 1]) {
                temp = a[j];
```

```

        a[j] = a[j + 1];
        a[j + 1] = temp;
    }
}
}
int main()
{
    clock_t start,end; double total_cputime;
    start=clock();
    int n;
    printf("Enter the value of n = ");
    scanf("%d",&n);
    printf("<-----BEST CASE----->\n");
    int arr[n];
    for(int i=0;i<n;++i)
        arr[i]=i;
    bubbleSort(arr,n);
    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);
    printf("<-----AVG CASE----->\n");
    start=clock();
    for(int i=0;i<n;++i)
        arr[i]=rand()%5000;
    bubbleSort(arr,n);
    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);

    printf("<-----Worst CASE----->\n");
    start=clock();
    for(int i=0;i<n;++i)
        arr[i]=n-i;
    bubbleSort(arr,n);
    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);
    return 0;
}

```

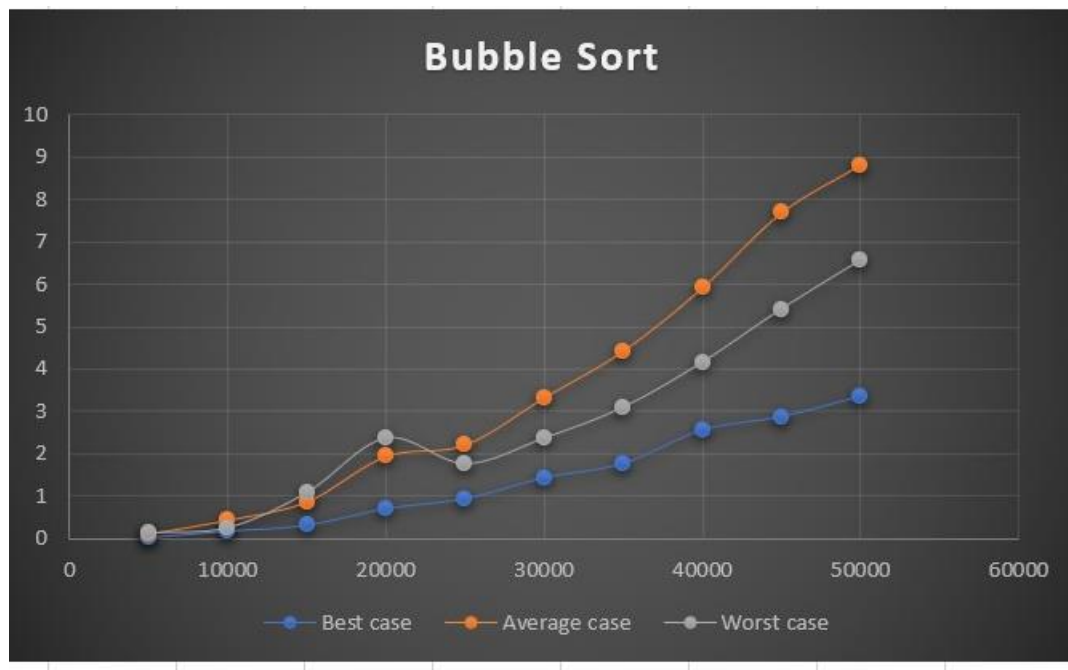
Output

```

<-----BEST CASE----->
total cpu time in sec = 0.041745
<-----AVG CASE----->
total cpu time in sec = 0.111071
<-----Worst CASE----->
total cpu time in sec = 0.150295

```

Graph



2. Write a C program for selection sort to

- I. Compare the time complexity with the given data set given below and calculate the time complexity based on the CPU clock.
- II. Plot a graph showing the comparison (n, the input data Vs. CPU times)

Sl No.	Value of n	Selection Sort (Time Complexity)		
		Best case	Average case	Worst case
1	5000	0.036660	0.037115	0.037249
2	10000	0.147260	0.147913	0.150601
3	15000	0.351986	0.344208	0.336222
4	20000	0.632585	0.630308	0.620012
5	25000	0.925247	0.925438	0.989290
6	30000	1.665745	1.416290	1.421739
7	35000	1.932051	1.800154	1.814272

8	40000	2.910240	2.602382	2.376459
9	45000	2.987375	2.968348	3.005760
10	50000	3.874391	3.655116	3.721206

Program

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[],int n)
{
    int i, j, min_idx;

    for (i = 0; i < n-1; i++)
    {
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
        swap(&arr[min_idx], &arr[i]);
    }
}

int main()
{
    clock_t start,end; double total_cputime;
    start=clock();
    int n;
    printf("Enter the value of n = ");
    scanf("%d",&n);
    printf("<-----BEST CASE----->\n");
    int arr[n];
    for(int i=0;i<n;++i)
        arr[i]=i;
    selectionSort(arr,n);
    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);
    printf("<-----AVG CASE----->\n");
    start=clock();
    for(int i=0;i<n;++i)
        arr[i]=rand()%5000;
    selectionSort(arr,n);
```

```

    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);

    printf("<-----Worst CASE----->\n");
    start=clock();
    for(int i=0;i<n;++i)
        arr[i]=n-i;
    selectionSort(arr,n);
    end=clock();
    total_cputime=((double)(end-start))/CLOCKS_PER_SEC;
    printf("total cpu time in sec = %f\n",total_cputime);
    return 0;
}

```

Output

```

Enter the value of n = 50000
<-----BEST CASE----->
total cpu time in sec = 3.874391
<-----AVG CASE----->
total cpu time in sec = 3.655116
<-----Worst CASE----->
total cpu time in sec = 3.721206

```

Graph

